DEPARTAMENTO DE INFORMÁTICA

PROYECTO FIN DE GRADO

MODIFICACIÓN DE ESTADOS DEL TELÉFONO USANDO TECNOLOGÍA NFC

Autor: Omar Azzahraoui Daoudi

Tutor: David Palomar Delgado

Título: Modificación de estados del teléfono usando la tecnología NFC

Autor: Omar Azzahraoui Daoudi

Director: David Palomar Delgado

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día ___ de _____ del 20____ en Colmenarejo, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO RESIDENTE

Agradecimientos

Querría expresar mis más sinceros agradecimientos a todas aquellas personas que me han concedido su apoyo incondicional a lo largo de mi carrera estudiantil, mi familia. También, agradecer a aquellos que han contribuido en la realización de este proyecto de Fin de Grado:

En primer lugar, a mi familia por su apoyo constante. En especial, agradecer a mis padres por la infinita paciencia que han tenido conmigo a lo largo de mi carrera como estudiante. También, agradecer a mi mujer por estar siempre a mi lado, y darme el empujón necesario para superar los obstáculos que se me han ido apareciendo.

Un agradecimiento especial a David Palomar, tutor de este proyecto, por sus consejos para afrontar las distintas dificultades que he tenido a lo largo de la realización del mismo.

Por último, y no menos importante, dar las gracias a mis compañeros de la Universidad por el gran ambiente de compañerismo que reinaba.

Resumen

En este proyecto se propone un uso más extendido de una tecnología novedosa que poco a poco van incorporando prácticamente todos los teléfonos móviles inteligentes. Se pretende hacer uso de **Near Field Communication** (en adelante **NFC**) para realizar algunas tareas cotidianas y repetitivas. Estas tareas consisten en cambiar los estados del teléfono simplemente acercando el móvil a una etiqueta NFC, llevar a cabo un control de presencia y control de acceso.

Para que sea posible proseguir con esta idea hace falta fundamentalmente un teléfono móvil con tecnología NFC incorporada y etiquetas NFC para almacenar los códigos de operaciones. También es importante mencionar que el teléfono móvil debe tener un sistema operativo Android. La razón fundamental por la que se ha elegido trabajar con Android es la cuota de mercado que tiene y, porque a día de hoy la mayoría de los teléfonos con Android llevan incorporados la tecnología NFC. Es más, es el único que ofrece las API necesarias para interaccionar con NFC con cierta libertad. Este trabajo se traduce en crear una aplicación móvil que constará básicamente de dos partes: una de ellas trabajará de forma local y otra precisará trabajar interconectada con un servidor.

El cambio de los estaos del teléfono consiste en acercar el móvil a una etiqueta NFC con un código específico que representa un modo y, el teléfono cambiará de modo. Un ejemplo de ello consiste en cambiar el modo del teléfono de silencio a normal y además se desactiva la conexión WIFI y se activan los datos móviles. Para poder hacer este cambio comentado, hay que disponer de un código que estará contenido en la etiqueta NFC, dicho código lo genera la propia aplicación y también se encargará de guardarlo, esta funcionalidad se denomina “definir modo”.

Para darle un uso más extendido a la tecnología NFC y para fomentar el uso del móvil en los entornos laborales, se van a desarrollar dos módulos totalmente enfocados a este entorno. Control de presencia: consiste en usar el teléfono móvil para registrar la asistencia al trabajo registrando para ello la fecha, hora y las coordenadas GPS de la posición actual del teléfono móvil. Esta acción, registrar la entrada y la salida del puesto de trabajo, es una práctica necesaria en la mayoría de las empresas por no decir todas. El

desarrollo de este módulo tiene como objetivo eliminar todo tipo de tarjetas, contraseñas, etc. que se usan actualmente, y fomentar el uso del teléfono móvil en el entorno laboral.

Control de acceso: acceder de forma exclusiva a ciertas secciones de un edificio es algo habitual en la mayoría de las empresas. El propósito de este módulo es sustituir los mecanismos existentes actualmente por el teléfono móvil, consiguiendo usar así el teléfono de una forma más activa y unificar todas las tarjetas o llaves en un solo teléfono móvil.

Índice general

Índice de figuras

Índice de tablas

Sección 1

1. Introducción y objetivos

1.1. Introducción

En esta sección de la memoria se abordará el proyecto desde una perspectiva más abstracta. Se hará un resumen general de toda la documentación recogida en ella. Se empezará describiendo el proyecto y los objetivos que se desea alcanzar con la realización del mismo. Se mencionará, aunque de forma resumida, la fase del desarrollo y se comentarán los medios (hardware y software) empleados. Para finalizar esta sección se describirá la estructura que sigue esta memoria.

1.2. Descripción del proyecto

El proyecto consiste en crear una aplicación móvil capaz de modificar los estados del teléfono de forma automática, basándose en información que obtiene por medio de etiquetas NFC. Cada etiqueta almacenará un código que hará referencia a un estado concreto. Para poder obtener esta información a partir de una etiqueta NFC será necesario guardarla previamente. Este proceso se define en la aplicación como definición de modos, que estará disponible como una de las opciones de la aplicación. El usuario, una vez seleccione la opción correspondiente a la definición de modos, elegirá el modo de audio, así como, la posibilidad de activar por ejemplo el módulo Wifi, Bluetooth, GPS, datos móviles, etc. El usuario, una vez tenga definido el modo, seleccionará la opción correspondiente para guardar estos datos en una etiqueta NFC acercando el dispositivo a la etiqueta.

Por lo tanto, con dicho proyecto se pretende facilitar la tarea de cambiar de estado del teléfono, haciendo que el usuario tenga que llevar a cabo una única tarea: acercar el móvil a una etiqueta NFC, en vez de atender varios pasos para conseguir finalmente aplicar el modo requerido.

Algunas propuestas del proyecto para hacer un uso más extendido de la tecnología NFC son: utilizar el móvil para registrar la entrada y salida del trabajo; y, usarlo como llavero para acceder a zonas restringidas o para abrir una caja fuerte. Esta última propuesta se presenta como dos módulos nuevos llamados "Control de presencia" y "Control de acceso".

El módulo Control de presencia identifica al personal de una empresa y registra la fecha y la hora en la base de datos. Así como, las coordenadas GPS de donde se encuentra en el momento de realizarse dicho registro. Este proceso es llevado a cabo en casi todas las empresas con el propósito de tener un mayor control sobre sus trabajadores. Los mecanismos usados hoy en día, tales como tarjetas de banda, tarjetas inteligentes, firma sobre papel, etc. son mecanismos poco fiables por el hecho de que se podría cometer fraude bien cediendo la tarjeta o firmando en lugar de otra persona. Sin embargo, el método que aquí se propone es fiable por varias razones: por una parte, el trabajador no puede realizar el fichaje a no ser que se encuentre muy cerca de la etiqueta NFC, la cual contiene la información necesaria para completar el proceso de fichaje. Por otra parte, es poco probable que alguien deje su dispositivo móvil a otros debido a la dependencia [28] que se tiene al teléfono móvil y puede contener información sensible y personal. No obstante, siempre existe el riesgo de que una persona ceda el dispositivo a otra para realizar dicho proceso. Para conseguir minimizar este riesgo vea el apartado de trabajo futuro.

La opción de "Control de acceso", consiste en usar el móvil como llave para acceder a zonas restringidas de un edificio o cualquier objeto guardado bajo llave. Esta llave consistirá en el envío de un código único al servidor y éste se encargará de abrir la puerta permitiendo acceso a la persona que aporte el móvil siempre y cuando tenga los permisos para ello. El proceso es similar al anterior, es decir, una persona acerca el móvil a una etiqueta NFC, la APP enviará la petición al servidor y éste responderá en función de si esta persona tiene o no acceso, además el servidor enviará una señal al mecanismo de la puerta para permitir acceso en caso de que se tengan permisos.

En general, el uso de tarjetas tiene el inconveniente de acumular en exceso las mismas llegando al punto de no disponer de sitio para más. E incluso, existe el riesgo de perder alguna con mucha facilidad. Por lo que, a través de este proyecto se propone unificar todo en uno a través del dispositivo móvil.

Los objetivos principales buscados en la realización de este proyecto son:

- Crear una aplicación capaz de modificar los estados del teléfono de forma automática y sólo basándose en la información contenida en las etiquetas NFC.
- La aplicación será capaz de guardar los modos definidos por los usuarios en etiquetas NFC.
- Con la pantalla del dispositivo desbloqueada, se podrá aplicar el modo que contenga una etiqueta de forma que no haga falta iniciar la aplicación.
- La aplicación permitirá realizar fichajes (registrando fecha, hora, coordenadas GPS y usuario que lo hace) sin necesidad de introducir pines, contraseñas, etc… con sólo leer una etiqueta habilitada para ese fin y conectándose de forma remota a un servidor.
- El uso del dispositivo como llave para acceder a zonas restringidas. Este proceso se hará de forma similar al del control de presencia o fichaje, con la diferencia de que en este caso, el servidor enviará información al dispositivo de la cerradura de la puerta para indicar o dar la orden para que se abra. En este proyecto se simulará este proceso por medio de un Arduino.

1.3. Objetivos personales

Para la realización de este proyecto, me he propuesto una serie de objetivos personales que describiré a continuación. Estos objetivos ayudarán a confeccionar este proyecto y, también aportarán conocimiento y experiencia a mi futura carrera profesional.

- Estudio en profundidad del lenguaje de JAVA.

- Los servicios web y en especial los basados en REST son claves para la realización de la parte de control de presencia y control de acceso. Para conseguir interconectar dos aplicaciones cuyas plataformas son distintas será necesario profundizar en el estudio de esta tecnología.
- Estudio y aprendizaje de la tecnología de comunicaciones de campos cercanos NFC, en la actualidad desconozco por completo el uso y la implementación de sistemas basados en ella, por lo que me he propuesto realizar una investigación teórico-práctica acerca de esta tecnología.
- Dado que la aplicación se va desarrollar para el sistema operativo Android y dado que no tengo experiencia en implementación de aplicaciones para estos sistemas, me he propuesto hacer un curso intensivo para poder llevar a cabo este proyecto.

1.4. Fase del desarrollo

Para el desarrollo del presente proyecto, se va a seguir una metodología basada en desarrollo en cascada con las fases de estudio de viabilidad, análisis y captura de requisitos, el diseño conceptual y de la arquitectura, implementación y por último se definen las pruebas. Para la realización de este proyecto se estima una duración de entre 3 hasta 5 meses, este proyecto se ha comenzado en abril de 2016 y se estima presentar en octubre del mismo año, la media de tiempo que se estima dedicar al proyecto es de aproximadamente unas quince horas semanales. A continuación se expone la duración de cada uno de los objetivos:

- La fase de estudio de viabilidad y reconocimiento del problema fue uno de los más largos y costosos del proyecto. Esto se debe a varias razones: una de estas razones ha sido el aprendizaje y la familiarización con la tecnología NFC así como la búsqueda de herramientas y tecnologías capaces de resolver el problema de forma eficiente. También ha sido muy costoso trabajar por primera vez con Android por falta de experiencia en este campo. Ésta tarea ha tenido una duración de un mes y dieciocho días (unas noventa y seis horas).

- La siguiente fase, análisis y captura de requisitos, ha tenido una duración de treinta y seis horas totales repartidas en doce días laborales. Durante esta fase, se han especificado los requisitos de software y se ha definido el alcance de la aplicación. También se han elaborado los documentos de los requisitos y los casos de uso.
- En la fase de diseño, se han elaborado los distintos diagramas a partir del documento de requisitos organizando así la información para que posteriormente resulte más fácil su implementación. Se han diseñado las arquitecturas física y lógica del sistema, así como el diseño tanto estático como dinámico de la aplicación y de la base de datos. Esta fase ha tenido una duración de cincuenta y tres horas (tres semanas y medio aproximadamente).
- En total, la fase de implementación tuvo una duración de un mes y ocho días unas (setenta y siete horas aproximadamente). Durante la fase del desarrollo se hicieron varios prototipos de diseño y, finalmente se escogió uno de ellos que fue la base para empezar a desarrollar la funcionalidad.
- Fase de pruebas. Aunque la fase de pruebas intensivas se ha realizado al final de proyecto, también se han realizado pruebas durante el desarrollo de la aplicación. El plan de pruebas se ha diseñado durante la fase de diseño, sin embargo, la ejecución de del plan de pruebas se ha realizado al final de la fase de desarrollo o implementación. Esta fase ha tenido una duración de veintiocho horas (unos nueve días aproximadamente). Esta fase tiene como objetivo asegurar que el software construido cumpla con los requisitos y, por tanto, verificar dichos requisitos contra la aplicación.

La documentación se ha elaborado conforme iba avanzado el proyecto. Parte de esta documentación era necesaria para la realización de las distintas fases del proyecto, como las tablas de requisitos, casos de uso o los diagramas de diseño. Sin embargo, la otra parte se realizó cuando se finalizó la fase de implementación y pruebas de la aplicación. La mayoría de esta documentación se ha reunido para la realización de esta memoria.

En total, el proyecto ha llevado unas setecientas noventa y cinco horas, nueve meses aproximadamente.

1.5. Medios empleados

Para desarrollar un proyecto software es necesario unos medios imprescindibles, a continuación se exponen los distintos medios empleados para la realización de éste en concreto. En este apartado, se expondrán únicamente los elementos hardware y software.

1.5.1. Hardware

- Hardware para el desarrollo del proyecto.
- Ordenador portátil para la realización de las distintas fases del proyecto, desde la toma de requisitos hasta las pruebas. También ha sido un elemento imprescindible para practicar todo aquello aprendido sobre Android.
- Móvil con tecnología NFC para ejecución de la aplicación y realización de las distintas pruebas.
- Etiquetas NFC para guardar la información que alimenta la aplicación.
- Arduino mega 260 para simular acceso a zonas restringidas.
- Diodo Led blanco para simular la apertura de la puerta.
- Resistencia para evitar quemar el Led.
- Protoboard para montar el sistema de simulación de puertas.
- Cableado para conectar el Led y la resistencia a la placa Protoboard.
- Hardware para las copias de seguridad.
- Disco duro externo Lacie de 300gb.
- Memoria USB PNY de 16gb.
- Hardware para la documentación.
- Impresora láser HP P1005 para la impresión de los documentos en papel.

1.5.2. Software

- Entorno de desarrollo.
- Eclipse.
- Arduino IDE.
- Servidores.

    - GlassFish Server. Se utiliza como servidor de aplicaciones para ejecutar los servicios web para el módulo de fichaje y acceso a zonas restringidas.
    - MySQL Server. Servidor de bases de datos donde se almacenan las tablas de las bases de datos de los módulos de fichaje y acceso a zonas restringidas.
- Librerías.
    - JDBC. Librería de MySQL para acceso a las bases de datos desde una aplicación.
    - Arduino. Librería no oficial desarrollada por investigadores de la Universidad de Panamá para conectar desde una aplicación java con un dispositivo Arduino.
- Software de documentación.
    - Microsoft Word 2011. Realización de la memoria y la presentación de algunas conclusiones de la misma.
    - Microsoft Visio. Se ha usado para realizar los diseños de los distintos diagramas.
    - Microsoft Project 2013. Se ha usado para realizar la planificación del proyecto.

1.6. Estructura de la memoria

Este documento incluye toda la información relacionada con la elaboración de este proyecto, y para facilitar su lectura se expone a continuación un resumen de cada sección contenida en la misma.

Sección 1: Introducción y objetivos.

Esta sección recoge una visión general de proyecto: incluye un resumen del mismo así como los objetivos a alcanzar tanto del proyecto como los personales, la metodología seguida y los medios empleados para su realización.

Sección 2: Estado del arte.

En esta sección se hablará del estado del mercado en el ámbito que se trata en este proyecto. Se analizarán las aplicaciones similares y posibles tecnologías a utilizar. Para concluir esta sección, se explicarán las tecnologías escogidas para la resolución del problema planteado.

Sección 3: Entorno de desarrollo.

Se describen las características de Hardware y Software que se han utilizado para el estudio, desarrollo y ejecución del proyecto, incluyendo la información sobre las versiones utilizadas. Además, se incluye una guía sobre los distintos pasos a seguir para la instalación y configuración del entorno de desarrollo.

Sección 4: Análisis del sistema.

Esta sección resume los casos de uso y requisitos de software recogidos en la fase de análisis. Los documentos completos se podrán consultar al final de la memoria.

Sección 5: Diseño de sistema.

En la sección 5 se recogen y explican los diagramas obtenidos en la fase de diseño. Se expondrá la arquitectura del sistema, el diseño estático y dinámico de la aplicación desarrollada y el diseño de la base de datos.

Sección 6: Pruebas.

El plan de pruebas es usado para verificar y comprobar el correcto funcionamiento del software desarrollado y también se usa para verificar la aplicación contra los requisitos de software acordados, todo esto se recogerá en la sección 6.

Sección 7: Conclusiones y trabajo futuro.

Se ha visto recomendable añadir esta sección en donde se recogen las conclusiones una vez finalizado el proyecto. Se expondrán las dificultades encontradas y posibles trabajos futuros así como, también, se analizarán los objetivos planteados al inicio y verificar si se han cumplido.

Sección 8: Presupuesto.

La sección 8 será dedicada al presupuesto del proyecto. Se expone la planificación y duración de cada una de las fases del proyecto. Se incluirá un diagrama de Gantt. También se expone el coste total, el cual se obtiene calculando los costes de personal, elementos hardware, software y los materiales fungibles.

Glosario.

El glosario recogerá los distintos acrónimos y términos poco comunes junto con su significado.

Referencias.

Se listan las fuentes bibliográficas consultadas a las que se hace referencia en la memoria

Apéndices.

Esta sección incluye los documentos de casos de uso y requisitos de software completos. Esta información se recoge también en la sección 4 de forma resumida.

Sección 2

2. Estado del arte

2.1. Introducción

El proyecto, como ya se ha explicado en la primera sección de esta memoria, consiste en una aplicación móvil capaz de modificar los estados del teléfono de forma automática, y sin necesidad de que el usuario tenga que elegir de forma manual en un determinado momento el modo que quiera aplicar. Esta funcionalidad estará disponible para todo el público, mientras otras funcionalidad como: control de presencia y control de acceso, se desea enfocar a unos usuarios más específicos.

Esta sección constará de un resumen general del problema que se desea resolver. Por un lado, presentando una visión general de las tecnologías similares, que se usan en la actualidad. Por otro lado, las aplicaciones disponibles en el mercado cuyo funcionamiento o enfoque sea similar a la que se pretende desarrollar.

Una vez abordado esto, se expondrá la solución adoptada al problema ya planteado en este proyecto, indicando las tecnologías que se pretende usar y explicándolas más detalladamente.

2.2. Planteamiento del problema

Desde la aparición del primer teléfono móvil, surgió la necesidad de silenciar el teléfono, no sólo quitarle el sonido sino que también se inventó el vibrador para que avisase de otra forma más discreta. De hecho, el vibrador lo inventó Motorola en 1984 [1], es decir, con la salida de los primeros teléfonos al mercado.

Con el auge de la telefonía móvil y, sobre todo, con la aparición de los teléfonos inteligentes o Smart Phone y la inclusión de tecnologías nuevas en el mismo, ha hecho necesaria su gestión y administración. Un teléfono de segunda generación (década de los noventa) ya permitía seleccionar entre varios modos de sonido, con los teléfonos de hoy en día, prácticamente son configurables al 100%.

Habilitar o deshabilitar el módulo Wifi del móvil por ejemplo, o simplemente cambiar el estado del teléfono a modo silencio, no es una tarea difícil de llevar a cabo. Sin embargo, si hay que combinar varias tareas y además se hacen repetidas veces durante un día resultan tareas tediosas. Además de cambiar el modo de sonido, a veces interesa también cambiar el modo de Wifi, Bluetooth, GPS… estas necesidades surgen para evitar molestar a los demás o simplemente para ahorrar la batería del móvil, ya que uno de los principales problemas de los teléfonos móviles de hoy en día es la poca autonomía de las baterías.

En los teléfonos más recientes, de tercera y cuarta generación, los fabricantes de los sistemas operativos y la comunidad de desarrolladores de software para móviles, llevan a cabo mejoras para hacer que estas tareas sean lo menos tediosas posibles, es decir, tener al alcance e intentar que con un solo click el teléfono cambie de estado de sonido y además apague los módulos Wifi y Bluetooth entre otros.

Las versiones más recientes de los sistemas operativos Android y IOS, permiten acceso desde la pantalla principal a un Widget, donde de una forma fácil y rápida se puede habilitar o deshabilitar las distintas funciones de las que dispone el terminal.

Hoy en día, el teléfono ha pasado de ser un dispositivo para llamar y enviar mensajes SMS exclusivamente, a ser un ordenador potente con capacidades que superan en ocasiones a algunos ordenadores de sobremesa o portátiles de hace unos años.

Llevar a cabo un control de presencia, también denominado “fichajes de entrada y salida” dentro de las empresas, o lo que es lo mismo: registrar la hora de entrada y salida al puesto de trabajo. Esto resulta casi imprescindible en la mayoría de ellas, y cada vez más se usan tecnologías avanzadas. En muchas empresas, esta operación puede hacer

variar significativamente el salario de un trabajador en caso de no hacerse correctamente, ya que en muchos casos se entiende como la falta de puntualidad por parte del trabajador y, por tanto, no cumplir con lo acordado con la empresa.

Dependiendo de la organización, uno se puede encontrar con varias formas de realizar dicho registro. En algunas de ellas todavía hace falta firmar un documento en presencia de otra persona responsable. Esta manera, tan antigua de registrar este acontecimiento, todavía persiste a pesar de los grandes avances tecnológicos que se ha alcanzado. Se lleva a cabo por el simple hecho de que la empresa entiende que es una forma segura, que tiene de controlar que el trabajador cumple con sus obligaciones. Las empresas que han avanzado en esta materia están usando tarjetas inteligentes, huellas dactilares o un código secreto junto al DNI de la persona que hace el fichaje. Todo ello con el objetivo de llevar a cabo un control exhaustivo sobre el cumplimiento de la jornada laboral de los trabajadores.

Al igual que la necesidad que tienen las organizaciones de realizar controles sobre el cumplimiento de sus trabajadores con la puntualidad, también surge la necesidad de reservar el derecho de acceso a ciertos accesos para personas con previa autorización. Estos sitios en la mayoría de los casos son sensibles por el material que contienen o la información que se almacena en ellos. Establecer acceso a determinadas zonas por grado de responsabilidad es una práctica habitual en muchas empresas y centros públicos.

El acceso a las distintas instalaciones de una organización, al igual que el fichaje de entrada y salida, es una de las prácticas habituales que surgen por la necesidad de controlar estas instalaciones con el objetivo de salvaguardar y custodiar todo aquello que se considera sensible. Y, por lo tanto, exige que las personas que puedan acceder sean previamente autorizadas. Las tarjetas inteligentes, llaves (las de toda la vida), porteros con código de acceso, huellas dactilares son algunas de las muchas formas que se utilizan para permitir o denegar acceso, sin embargo, al igual que el caso de los fichajes que hemos visto antes, requiere llevar encima tarjetas, llaves, recordarse del pin, etcétera.

Como se ha visto ya en apartados anteriores, cambiar el estado del teléfono puede llegar a ser una tarea tediosa. Una forma de solucionar este problema es hacer que la tarea se haga de forma automática por medio de algún evento o algo similar.

Fichar la entrada o salida en el trabajo es una tarea diaria y prácticamente obligatoria en las empresas. Igualmente, acceder a una zona privada o restringida se hace por medio de un pin, contraseña, tarjeta inteligente o medio similar. Para llevar a cabo estas tareas, imagínese que es necesaria una tarjeta para cada tarea además de todas aquellas que ya se aportan en la cartera: como las tarjetas del banco, las de los puntos de distintas gasolineras o centros comerciales, DNI, tarjeta sanitaria, carnet de conducir, etc…

Fundamentalmente, por este motivo se plantea este proyecto: usar el móvil para tareas cotidianas desde cambiar los estados del mismo de una forma amena hasta sustituirlo por las típicas tarjetas o aparatos que son muy costosos de instalar.

2.3. Propuesta de solución

Para proponer una solución al problema mencionado anteriormente, se expondrán las tecnologías utilizadas así como el análisis previo que se hizo para que finalmente resultasen elegidas.

En un primer momento, pensando las posibles soluciones para el problema, empecé buscando información acerca de la tecnología NFC, tras este pequeño estudio e investigación, llegué a la conclusión de que no se puede realizar excepto para los dispositivos Android. Esta conclusión se dio ya que en primera instancia únicamente pensé en sistemas operativos Android y IOS, sin embargo, he descartado por completo IOS dado que los únicos terminales que llevan incorporado la tecnología NFC son el iPhone 6 y el 6s y, además, el sistema operativo IOS no ofrece las API necesarias para interaccionar con dicha tecnología, Apple sólo permite el uso de NFC para micro-pagos (aplicación desarrollada por Apple incorporada en IOS 8.1 o superior en Estados Unidos

y 8.3 o superior en Inglaterra [2]) esta decisión, se ha tomado sobre todo para prevenir riesgos de seguridad [3]. Android por su parte, ofrece las API para usar NFC [4] con ciertas restricciones para prevenir, al igual que en el caso de Apple, los altos de seguridad.

Una vez elegida la plataforma, me he puesto a estudiar siguiendo el libro **El gran libro de Android avanzado** [5]. En un principio, lo vi como una desventaja ya que es un lenguaje que no conozco y por lo tanto me temía lo peor, sin embargo, todo lo contrario. Android, aunque se basa en Java fundamentalmente, pero también tiene librerías que hacen más fácil ciertas tareas cotidianas de desarrollo y, las cuales conviene conocer y saber utilizar. De verlo como desventaja pasó a una ventaja enorme de cara a mi futuro profesional, esto es porque es un lenguaje que se usa mucho en las empresas que se dedican a desarrollar las aplicaciones móviles, una de las razones de la demanda de programación para Android, es la cuota de mercado que tiene asignada.

Una de las cosas que más me ha costado conseguir fue programar lectura y escritura de etiquetas NFC, una vez alcanzado este objetivo, todo fue más o menos fácil de conseguir ya que el resto de la dificultad residía únicamente en la lógica de negocio de la aplicación.

Una vez asentada la idea a desarrollar y conocida la plataforma y el lenguaje de programación, me dediqué a pensar el diseño de la aplicación basándome en aplicaciones ya hechas e ideas que el tutor me había sugerido.

Una de las ideas que tenía para la aplicación a desarrollar era la posibilidad de que la aplicación funcione en segundo plano como un servicio a la espera de recibir el evento al leer una etiqueta, sin embargo, no es posible por las limitaciones que impone Android por temas relacionados con la seguridad, ya que al igual que Apple, Android, también usa la tecnología NFC para realizar micro-pagos y por ello, como ya veremos, nos enfrentaremos a muchas limitaciones y todas ellas serán por esta causa.

Para poder aplicar esta solución usando NFC y como ya he mencionado anteriormente, necesitamos un elemento externo al teléfono que contenga información acerca del modo a usar o información de la empresa para realizar el fichaje o abrir una puerta, este elemento lo he mencionado anteriormente, es la etiqueta NFC. La etiqueta contendrá la información necesaria codificada de forma que, al ser excitada por el dispositivo con la aplicación instalada ejecutará la opción correspondiente.

Cambiar los estados del teléfono es una operación llevada a cabo de forma local, es decir, se procesa dentro del teléfono y la única información externa que se necesita es la contenida en la etiqueta NFC. Fichar, sin embargo, es necesario procesar la información contenida en la etiqueta además de la información contenida en la aplicación. El dispositivo lee el contenido de una etiqueta, ésta contiene información codificada de la empresa, como es en este caso el CIF, la aplicación recoge este dato y lo envía junto con el código único generado durante el proceso de autenticación a un servidor donde se comprueba que dicho código (único para cada dispositivo enlazado) sea un código válido y además se encuentra en ese momento asignado a una persona, que a su vez está relacionada con la empresa cuya identificación se encontraba en la etiqueta NFC.

Una vez comprobados los datos, el servidor almacenará la información de fichaje en la base de datos de forma permanente. Para hacer posible este tipo de arquitectura, he optado por usar un servicio Web basado en REST, de forma que la aplicación se pondrá en contacto con el servicio Web, le entregará la información tal como se ha descrito arriba y el servicio Web se encargará de realizar la búsqueda en la base de datos y comprobar que los datos sean correctos, y enviará de vuelta un código de operación que indicará si se ha realizado correctamente o no.

Para la parte de llavero, se usará el mismo sistema con la diferencia de que el componente Web, antes de devolver un código de operación que indique que los datos son correctos, enviará una señal al sistema de apertura de puertas para abrirlas.

2.4. Tecnologías

2.4.1. NFC

2.4.1.1. Introducción

La tecnología conocida como Near Field Communication o NFC puede considerarse como un avance del RFID o Radio Frequency Identification [6]. Básicamente, NFC permite realizar una comunicación simple, segura e intuitiva entre dispositivos.

NFC aparece como una evolución en el uso de aplicaciones dentro del teléfono móvil, pues se presenta como un sistema que comunicación sencilla, una alternativa para el manejo de pagos y una opción para el almacenamiento de datos de forma más segura para los dispositivos electrónicos móviles [7].

La principal característica que hace que la tecnología NFC sea interesante y atractiva, es que complementa a otras tecnologías inalámbricas como el Bluetooth, Wifi y el mismo RFID. La principal diferencia entre NFC y los otros esquemas sin contacto, es que no está pensada para la transmisión masiva de datos, pero sí para un intercambio casi instantáneo de una poca cantidad de información y no necesita un emparejamiento previo [7].

Para la mejor comprensión de la tecnología NFC, se debe conocer antes los principios de su antepasado, el sistema RFID.

2.4.1.2. El sistema RFID

Funcionamiento y componentes.

El Radio Frequency Identication (RFID) ya empezaba a tomar forma durante la Segunda Guerra Mundial, en donde se utilizaba la identificación por radiofrecuencia de manera masiva por los británicos para distinguir entre aeronaves propias o enemigas [8]. Actualmente RFID, ya como una tecnología bien constituida, puede definirse como aquel sistema que tiene como principal función la identificación de determinados objetos a

distancia, utilizando para tal efecto las ondas de radio. Para conseguirlo, esta tecnología proporciona soporte para el almacenamiento y la recuperación remota de datos en etiquetas o tarjetas RFID que contienen la información necesaria para el reconocimiento [9].

El funcionamiento de los sistemas de RFID es sencillo. Básicamente consta de tres partes como se ve en la Figura 1. Existe un lector RFID que de manera periódica busca en su zona de alcance la información contenida en las señales que son emitidas por alguna etiqueta RFID [9]. Estas etiquetas poseen la capacidad de adherirse a productos, personas o animales que necesitan ser identificados o seguidos [10]. Una vez que se hayan recibido los datos, el lector los transfiere a un subsistema de procesamiento para la interpretación y el tratamiento correspondiente [9].

Figura 1: Funcionamiento del sistema RFID [12].

Considerando las Etiquetas RFID, se puede decir que se componen principalmente por una antena, un transductor de radio y un material encapsulado o chip [13]. Se pueden clasificar las etiquetas tomando en cuenta la fuente de alimentación, considerando esto, existen básicamente tres tipos [14]:

- ***Etiquetas pasivas:*** Son las más económicas. No poseen fuente de alimentación propia, por lo que utilizan la pequeña corriente eléctrica inducida por la energía que reciben del lector, misma que es suficiente para poner en marcha su circuito y para realizar la transmisión de datos.

- ***Etiquetas semi-pasivas:*** Reciben parte de su energía de una fuente propia de alimentación. Esta batería les sirve para alimentar sólo sus circuitos, para la transmisión de información utilizan la energía del lector.
- ***Etiquetas activas***: Son más caras que las anteriores pero poseen mayor capacidad de almacenamiento. Además cuentan con su propia fuente de alimentación tanto para sus circuitos, como para la transmisión de la información.

Ventajas. La tecnología RFID presenta diversos beneficios y puede ser utilizada en innumerables aplicaciones. A continuación se detallan sus ventajas [12] [15]:

- Es un sistema bastante robusto y posee un buen desempeño en ambientes severos (lugares húmedos, sucios, y con variaciones de temperatura).
- No es necesario el contacto visual entre la etiqueta y el lector para la transferencia de datos o la comunicación.
- Un lector RFID puede leer múltiples etiquetas de forma simultánea.
- Las etiquetas RFID son resistentes y duraderas debido a que es posible insertarlas en materiales robustos.
- Con la tecnología RFID se puede brindar a un producto una identidad propia dentro de una línea de producción, la capacidad de comunicarse con su ambiente y la habilidad de conservar y obtener la información de sí mismo.

Problemas. Todos los esquemas de comunicación poseen ciertas desventajas, y el RFID no es una excepción. En las siguientes líneas, se mencionan los inconvenientes propios de este sistema [15] [11]:

- RFID no cuenta con un estándar internacional referente al uso de frecuencias, esto ocasiona problemas de incompatibilidad en sistemas utilizados entre algunos países y otros.
- El punto débil del RFID, es que no provee de mecanismos de seguridad para proteger la información ante las amenazas de ataques a la privacidad.
- La capacidad de almacenamiento y el procesamiento en las etiquetas RFID son insuficientes para implementaciones de mecanismos de seguridad tales como

técnicas criptográficas. Este hecho contribuye a que se complique el establecimiento de conexiones seguras para evitar que los datos sean captados y leídos por intrusos.

- Entre el lector y las etiquetas puede ocurrir una saturación de etiquetas, debido a la excesiva lectura de etiquetas al mismo tiempo, o la redundancia en la lectura, que se produce cuando una etiqueta es detectada por varios lectores.
- Otra posible interferencia en el sistema es lo que se conoce como escudo electromagnético, que es el efecto producido cuando un material conductor se encuentra posicionado entre una etiqueta y un lector, por ejemplo, envolver con papel de aluminio una etiqueta RFID.
- Con respecto a la seguridad, otra falencia es que la etiqueta puede ser leída a cierta distancia sin que el usuario tenga conocimiento de este hecho.
- El costo de la implementación del RFID en comparación con el sistema de código de barras es el principal factor limitante para su instalación, aunque con los desarrollos y mejoras, el uso de la tecnología RFID sería cada vez más económico.

Aplicaciones. La tecnología RFID posee aplicaciones muy variadas (Figura 2), como las que se mencionan a continuación [14] [16]:

- **Identificación y control**: En esta categoría se pueden encontrar cuatro grupos:
    - **Control de animales:** Se utilizan chips pasivos incrustados en el animal a nivel subcutáneo que permite su identificación en caso de extravío. Para el control de ganado se utilizan crotales.
    - **Control de acceso:** Se encarga de controlar el acceso de funcionarios a los edificios, además de registrar y gestionar los horarios.
    - **Seguimiento de pacientes:** Se trata de una pulsera que permite localizar al paciente dentro del hospital y también permite acceder a su historial.
    - **Logística:** Una de las aplicaciones más comunes de la tecnología RFID es el control de inventario y seguimiento de artículos para agilizar los procesos contables y optimizar la monitorización de productos. Grandes empresas como Coca-Cola y McDonalds, han incorporado RFID en sus sistemas de distribución y almacenamiento de mercaderías. También

BMW y Toyota, la han utilizado para el control de procesos de manufacturas.

- **Sistemas de pago:** RFID se presenta como una opción para sistemas de pago, como por ejemplo la compra del billete del autobús urbano. En Salamanca se utilizan etiquetas RFID en tarjetas plásticas para almacenar el saldo del usuario. Cada vez que el cliente pasa la tarjeta por uno de los lectores en el autobús, el precio del billete correspondiente es descontado de su cuenta en la tarjeta. Sin embargo, un punto débil es que este sistema no ofrece medidas de seguridad ni autenticación para los pasajeros. Otro ambiente en donde se utiliza el RFID es para el control y cobro de peajes.

Figura 2. ALGUNAS aplicaciones del RFID. 1- Control de ganado. 2- Seguimiento de pacientes. 3- Logística. 4- Tele-peaje. [14].

2.4.1.3. La evolución hacia el NFC

El Near Field Communication brinda nuevas funciones a la tecnología RFID.
Esto se debe a la combinación de un lector y una etiqueta RFID en un mismo equipo NFC. De esta manera se facilita la comunicación en ambas direcciones entre dos dispositivos y se quiebra la separación funcional de los esquemas RFID, es decir, el lector por un lado y las etiquetas por el otro [17].

Es necesario considerar que la tecnología NFC tiene algunas diferencias con RFID, a pesar de que ambas utilizan el mismo tipo de chip y que una deriva de la otra. Lo que

debe quedar claro es que aunque NFC es un nuevo estándar que aparece a raíz de RFID, su misión no es reemplazarlo, más bien, complementarlo, aumentando sus funciones [14].

Realizando una comparación entre NFC y RFID, encontramos que la primera posee la capacidad de cómputo necesaria para ejecutar operaciones, hecho que hace fácil su integración en dispositivos como teléfonos móviles [14]. Además NFC provee una comunicación peer to peer, medio que permite intercambiar información entre dos dispositivos [13].

Otra cuestión que tiene NFC es que permite rangos pequeños de comunicación, y por ende posee una seguridad inherente, punto que lo hace preferible para cuestiones de comunicación que requieren seguridad como los medios de pago o intercambio de información personal. Además NFC no puede ser activado de forma remota, involuntariamente o por accidente. El teléfono obliga a que deba existir un acercamiento entre dispositivos antes de iniciar una comunicación [13].

Con todo esto, cabe mencionar que se hace necesaria la evolución hacia el NFC para aprovechar la robusta seguridad, la versatilidad por el hecho de que puede utilizarse en dispositivos móviles, la facilidad en el uso y la posibilidad de servir como una tecnología que brinde al usuario información útil y le permita interactuar con su ambiente.

2.4.1.4. Aspectos generales

A continuación algunos apartados de introducción a los sistemas NFC.

2.4.1.4.1. Definición del NFC

Near Field Communication (NFC), es una tecnología de comunicación de corto alcance, que permite el intercambio de datos entre dos dispositivos de manera inalámbrica. Es compatible con infraestructuras RFID, dado a que es un derivado del mismo [10].
El NFC fue creado en el año 2002 como un proyecto encabezado por Nokia, Philips y Sony, mismas empresas que componen la asociación NFC Fórum, para promover la utilización de este sistema en dispositivos móviles. En el 2003 fue aprobada como estándar ISO/IEC [18].

El sistema de corto alcance se compone de dos elementos: un iniciador y un objetivo, en donde cualquier dispositivo con NFC puede adoptar las funciones o el comportamiento de una de estas partes. El NFC puede ser instalado en cámaras fotográficas, reproductores, televisores, teléfonos celulares y hasta en controles remotos [11].

2.4.1.4.2. NFC Forum

En el año 2004, se constituye el NFC Forum, teniéndose como una organización sin fines de lucro que busca promocionar la utilización del NFC en dispositivos a través del desarrollo ciertas de especiaciones que intentan unificar los sistemas NFC [10].
El NFC Fórum fue creado por las tres entidades pioneras en el NFC, Philips, Sony y Nokia, y ahora ya cuenta con más de una centena de miembros, entre ellos, empresas del rubro tecnológico, organismos económicos y otras organizaciones sin fines de lucro [19]. Esta asociación fomenta el desarrollo de un ambiente en donde se consigan que las aplicaciones para el NFC sean seguras y puedan funcionar unas con otras sin problema. Para lograr esto se definen especificaciones tanto para la arquitectura de los sistemas, como los protocolos para lograr la operatividad en conjunto [19].

Figura 3. NFC Fórum [14].

2.4.1.5. Características

El NFC proporciona la comunicación inalámbrica de corto alcance mediante un campo magnético que permite el intercambio de datos, y opera en un ambiente en donde los dispositivos se encuentran separados una distancia de 20 cm como máximo. El sistema se maneja en la banda de frecuencia no licenciada de fc = 13.56 MHz, y un ancho de banda que oscila 7 KHz a cada lado de fc. Las comunicaciones pueden ser half o full duplex. Se utiliza el esquema de modulación Amplitud Shift Keying (ASK) y la codificación Manchester. Se disponen de tres velocidades de transmisión 106, 212 y 404 kbps que son fijadas por el dispositivo que inicia la conexión [18].

Un dispositivo NFC que comienza la comunicación y controla el intercambio de información es conocido como iniciador (similar al lector RFID), y el que responde al iniciador es conocido como objetivo. La comunicación puede realizarse en modo activo o en modo pasivo [20].

2.4.1.6. Estandarización

Existen diferentes estándares para el NFC, como los definidos por ISO/IEC (International Organization for Standardization/ International Electrotechnical Comision), el ETSI (European Telecommunications Standards Institute) y también el ECMA (European Computer Manufacturers Association). Estos especifican aspectos importantes en los sistemas NFC como la tasa de transferencia, el esquema para la codificación y modulación así como otros parámetros [10].

El ECMA-340 (ISO/IEC 18092) define la interfaz y modo de operación (NFCIP-1). El estándar ECMA- 352 (ISO/IEC 21481) define una segunda versión de la interfaz y modo de operación en NFC (NFCIP-2). Además, el ECMA-356 (ISO/IEC 22536) y el ECMA-362 (ISO/IEC DIS 23917) definen la interfaz RF y el protocolo de comunicaciones [18].

2.4.1.7. Modos de operación

Teniendo en cuenta el modo de operación el NFCIP-1 y el NFCIP-2 son los protocolos más significativos. A continuación se describen brevemente [6]:

- **NFCIP-1**: Combina dos protocolos de comunicación que pertenecen al RFID, tales como el MIFARE y el FeliCa [6], e incluye en ellos nuevos protocolos de transporte.

- **NFCIP-2**: Hace posible la combinación del NFC con lectores RFID logrando así una compatibilidad.

Como se mencionó anteriormente, para establecer una comunicación existen dos modos [18]:

- **Modo pasivo**: Debe existir un dispositivo que reciba y otro que emita, este último dispone de fuente eléctrica propia para funcionar, y debe generar una señal para el intercambio de datos. Por el otro lado el dispositivo receptor no posee baterías y debe aprovechar el campo incidente del emisor para el funcionamiento de sus circuitos.

- **Modo Activo**: Los dispositivos poseen energía propia, por lo que ambos son capaces de generar el campo electromagnético para la transferencia de datos.

Para los dispositivos NFC es posible hacer la comunicación con el otro par, actuando como etiqueta o haciendo de lector/escritor. Referente a esto, el NFC Forum define los siguientes modos de operación [6] (Figura 4):

- **Peer to peer**: Es utilizado cuando surge la necesidad de transmitir una reducida cantidad de datos (unos pocos kilobytes). Si se desea elevar la cantidad de datos en la transmisión, la tecnología NFC es utilizada para concretar una conexión inalámbrica con el soporte necesario para la comunicación, como por ejemplo Bluetooth [17].

- **Lectura/escritura**: En este modo, se tiene la capacidad de leer y escribir las etiquetas [17]. El dispositivo puede leer cuatro tipos de etiquetas, mismas que fueron definidas en el NFC Forum. Una vez establecida la comunicación es posible el intercambio de texto (en pequeñas cantidades), una dirección de internet o un número de teléfono [22].

- **Emulación de tarjeta inteligente**: Un lector puede identificar a un dispositivo NFC, como si este fuera una etiqueta NFC o una tarjeta inteligente. Este modo puede ser usado para medios de pago [17], transacciones bancarias, pagos rápidos y control de acceso [22].

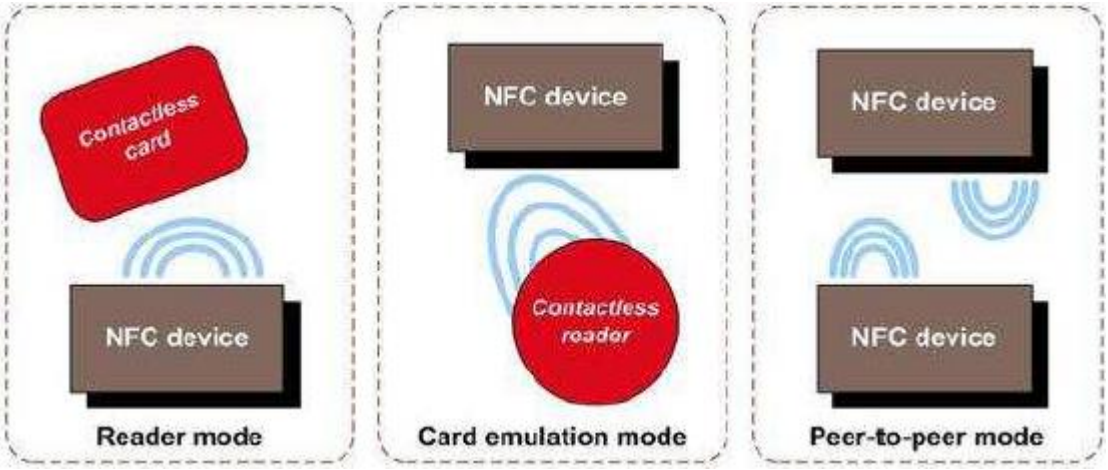


Figura 4 Modos de operación del NFC [13].

2.4.1.8. Arquitectura de un dispositivo NFC

En un dispositivo móvil NFC (Figura 5), es posible distinguir entre dos componentes fundamentales [14]:

- **Chip NFC y antena**: Este conjunto permite la comunicación y el intercambio de datos entre los sistemas NFC a muy poca distancia, por medio de un campo magnético. El chip se encuentra conectado al controlador banda base del teléfono, que es el encargado de la comunicación del móvil.

- **Elemento seguro (SE)**: Es un chip independiente que contiene las aplicaciones basadas en claves de seguridad, tiene como propósito permitir las transacciones seguras. Existen varias implementaciones para este elemento y se diferencian en su ubicación dentro del móvil.

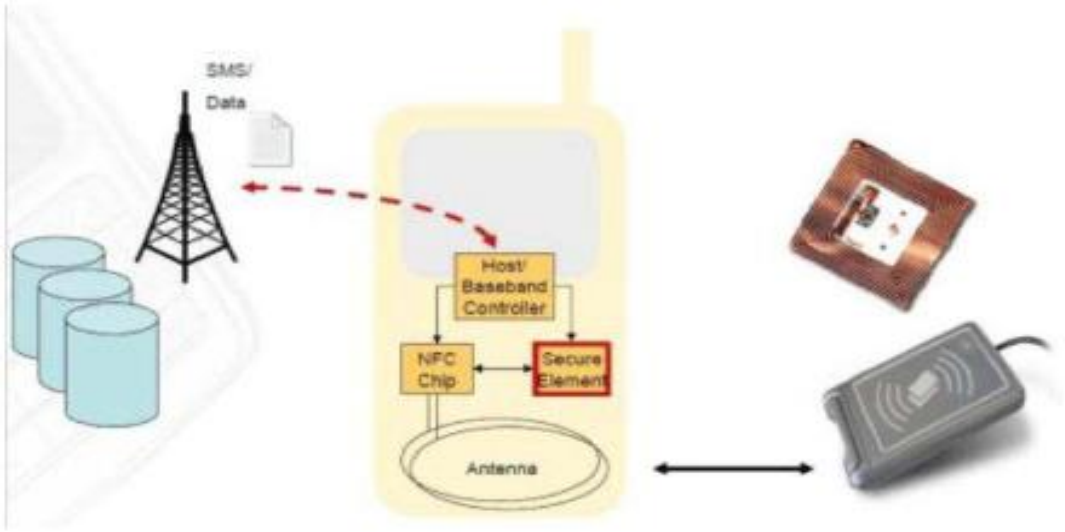


Figura 5. ARQUITECTURA NFC EN DISPOSITIVOS [14].

A continuación se detallan las implementaciones para el elemento seguro [13] (Figura 6):

- **SE incorporado en el circuito del móvil**: Es la arquitectura más utilizada en los proyectos a nivel mundial. En este caso el SE puede ser un chip ya montado en la placa base o conectado a ella de alguna manera. Su ventaja principal es que ya posee todas las certificaciones hardware y software necesarios, sin embargo, este modelo acarrea todo un problema cuando el usuario quiera cambiar de teléfono y deba de gestionar las credenciales de pago.

- **Tarjeta de memoria utilizada como SE**: En esta implementación una tarjeta de memoria incorpora un chip con un microcontrolador y una memoria flash.

- **Tarjeta SIM como SE**: Esta solución es más llamativa para las operadoras, porque de esta forma toda la gestión de la información estaría a su cargo. En este modelo, la tarjeta SIM incorpora la aplicación de pago, dicha aplicación puede cargarse en la propia tarjeta SIM.

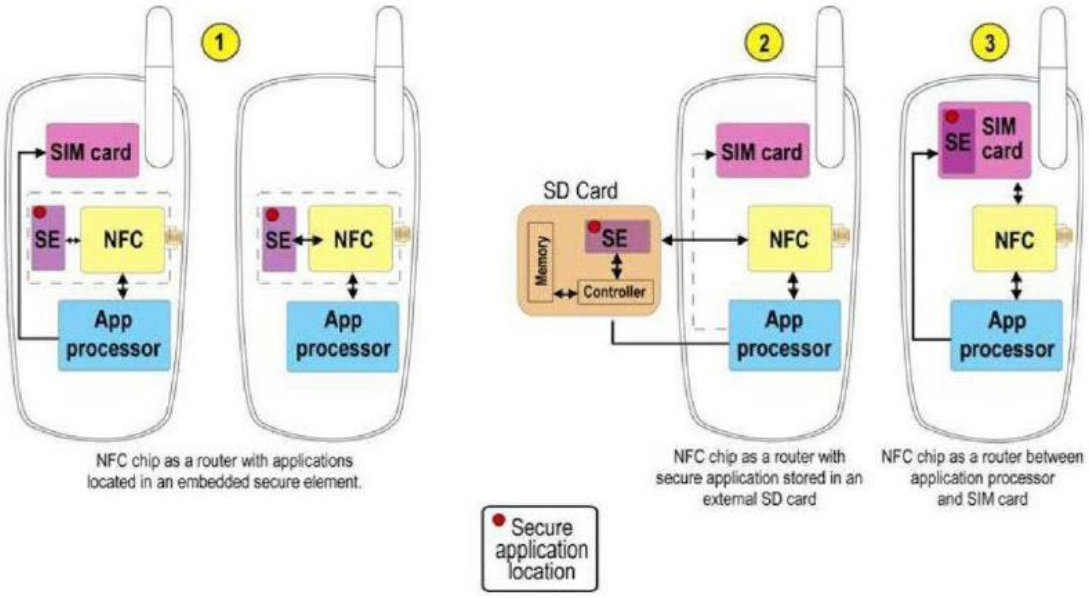


Figura 6. ELEMENTO SEGURO Y LOS MODOS DE IMPLEMENTACIÓN [13].

2.4.1.9. Etiquetas NFC

Constituyen una parte importante de la tecnología NFC, implementan un almacenamiento pasivo en la espera de que algún lector NFC requiera la información que retienen. El NFC Forum ha definido cuatro tipos de etiquetas [6].

En las especificaciones de las etiquetas se establecen las características de cada uno de los cuatro tipos, de manera tal a lograr la compatibilidad y operabilidad de los dispositivos en sus diferentes modos de lectura o escritura. En estas especificaciones se fijan varios parámetros [22] como los que se muestran a continuación en la Tabla 1.

2.4.1.10. Establecimiento de la comunicación NFC

En los sistemas NFC se pueden distinguir cinco etapas importantes que están presentes durante el establecimiento de la transacción. Estas fases son [22]:

- **Descubrimiento**: En esta fase inicial, los dispositivos se rastrean mutuamente y luego inician el reconocimiento.

- **Autenticación**: Cada uno de los dispositivos verifica si en el otro extremo su par está autorizado o si deben establecer una conexión segura a través de un cifrado correspondiente.

	Memoria	Especificaciones	Velocidad	Lectura/Escritura
Tipo 1	96 Bytes hasta 2 KBytes	ISO – 14443 A	106 kbits/s	Sí
Tipo 2	48 Bytes hasta 2 KBytes	ISO – 14443 A	106 kbits/s	Sí
Tipo 3	Hasta 1 MB	FeliCa ISO 18092	212 KBITS/S Y 424 KBITS/S [21]	Pre-configurados de fábrica como de lectura y escritura o sólo lectura.
Tipo 4	32 KBytes	ISO – 14443 A y B	106 kbits/s y 424 kbits/s	Pre-configurados de fábrica

Tabla 1. Tabla comparativa. Tipos de etiquetas NFC.

Negociación: Hasta este punto se definen parámetros como la tasa de transmisión, la identidad del dispositivo, la aplicación, y si es el caso, también la acción que van a solicitar.

Transferencia: En esta fase ya puede realizarse el intercambio de datos.

Confirmación: El receptor confirma el establecimiento de la comunicación y la transferencia de los datos.

Un aspecto que no debe pasarse por alto durante las transacciones, es la seguridad. Teniendo en cuenta esto, es posible utilizar un cifrado AES y triple DES para emular la protección que ofrece una tarjeta bancaria inteligente [14].

2.4.1.11. Formato de datos

Para que las etiquetas y los dispositivos puedan comunicarse entre sí [7], y se pueda conseguir la compatibilidad entre dispositivos NFC y RFID de los diferentes fabricantes [6], el NFC Forum define un formato de datos estandarizado.

2.4.1.11.1. NFC Data Exchange Format NDEF [23]

Se define un formato de encapsulación de mensaje para intercambiar información entre dispositivos NFC, ya sea de un dispositivo a una etiqueta o entre dos dispositivos NFC activos, también se especifican las reglas para construir un mensaje NDEF correcto, así como una cadena ordenada de registros NDEF.

NDEF no hace referencia a ningún circuito, ni arquitectura de conexión, ni se debe pensar que especifica el intercambio de información, es solamente un formato de mensaje. Este formato es el mismo para tarjetas, así como para dispositivos NFC, de esto se concluye que la información de NDEF no guarda relación con el tipo de dispositivo que participa en una comunicación.

Con este formato pueden transmitirse varios tipos de información, como:

- Documentos o fragmentos XML, imágenes de diverso formato y datos encriptados.
- Cadenas de información encapsulada.
- Documentos múltiples que guardan alguna relación lógica.

2.4.1.11.2. Record Type Definition RTD

Proporciona las pautas para la especificación de los tipos de registros, que pueden ser incluidos en mensaje NDEF. Esta especificación soporta aplicaciones específicas NFC [23].

El NFC Forum define dos tipos: NFC Forum External Types y NFC Forum Well-Known Types, siendo el primero creado para dar a otros organismos la posibilidad de especificar sus propios tipos de forma independiente [6].

Con respecto al NFC Forum Well-Known Types, es necesario decir que fue estandarizado por las especificaciones del NFC Forum, que proporcionan la pauta para el procesamiento y representación de los datos. Ellos son [6]:

- **Text Record Type:** Sólo texto simple, ninguna aplicación específica asignada.
- **Uniform Resource Identifier (URI) Record Type:** Correo electrónico, direcciones de Internet, números de teléfono u otros códigos de identificación.
- **Smart Poster Record Type:** Es una extensión del tipo de registro URI, que proporciona información adicional acerca del URI, como iconos o acciones recomendadas.
- **Generic Control Record Type:** Proporciona una estructura para cualquier actividad de control.
- **Signature Record Type:** Una firma está prevista para certificar la veracidad de los datos.
- **Connection Handover:** Ofrece traspaso de una conexión NFC a otra tecnología de comunicación con mayor rendimiento de datos (por ejemplo, Bluetooth).

2.4.1.12. NFC y otras tecnologías inalámbricas

En esta sección se presentan las diversas características de otros sistemas inalámbricos de comunicación con la intención de realizar una comparación con la tecnología NFC.

Lo que se pretende también es la enumeración de los diversos aspectos del Near Field Communication que se han ido mencionando hasta este punto del documento. Se detallan a continuación [14] [23]:

- Debido a la corta distancia en la que trabaja, NFC posee una cierta seguridad, pues sólo se realiza la operación correspondiente si el usuario lo solicita acercando su móvil NFC a otro dispositivo para establecer la comunicación.
- NFC facilita la utilización de las otras tecnologías, como Bluetooth o WIFI.

- Ya está preparada para la seguridad que requieren las aplicaciones externas, debido al soporte que ofrece para la protección.
- No se necesita conocer a fondo el funcionamiento del dispositivo, y podría denominarse como tecnología intuitiva, ofreciendo sus beneficios a un mayor número de personas.
- Es una tecnología abierta y basada en estándares ISO, ECMA y ETSI.
- Es versátil, debido a que puede ser utilizado en un gran número de sitios con diversas aplicaciones.
- Es una tecnología inalámbrica que opera en la banda ISM (Industrial, Scientific and Medical), que no necesita licencia y está mundialmente disponible.
- Posee gran alcance y disponibilidad, teniendo en cuenta que puede implementarse en cualquier teléfono móvil que no necesariamente debe ser de última generación.
- Facilita la migración al modo electrónico, pues permite funciones de pago, o controlar el acceso a lugares sin la necesidad de llaves.

Ahora que ya se han rememorado las características más importantes del sistema NFC, se procede a comparar al NFC con sus pares más conocidos en la Tabla 2.

	NFC	Bluetooth	ZigBee	IrDA	RFID
Tiempo de establecimiento	< 0,1s	6s	30ms	0,5s	< 0,1s
Velocidad de transmisión	424 Kbps	2.1 Mbps (Versión 3.0: 24 Mbps)	250 Kbps	2400 bps a 4 Mbps	424 Kbps
Alcance	10 cm	30 cm	70 cm	1 m	3 m
Seguridad	Bastante alta	Alta con PIN	Alta	Visión directa	Buena
Facilidad de uso (conexión)	Muy fácil (un toque)	Configuración, emparejamiento	Sin configuración	Sin configuración	Sin configuración

Usos	Acceso edificios, pagos, obtener información, establece conexiones, etc.	Red para intercambio de datos variados (audio, imágenes, documentos)	Domótica, control industrial, monitorización pacientes	Intercambio de datos y control remoto de dispositivos	Seguimiento, identificación de productos, etc

Tabla 2. Tabla comparativa. Tecnologías inalámbricas [14].

Entre las características citadas, se puede observar que NFC ofrece un buen nivel de seguridad, es de fácil uso, y el sistema puede utilizarse en las actividades diarias.

2.4.1.13. Medidas de Seguridad en NFC [24]

Para asegurar y proteger al usuario se deben tener en cuenta estos aspectos:

Anonimato: No debería ser posible la identificación de una etiqueta/token NFC.

Privacidad de la localización: No deber¶³a ser posible la trazabilidad de la localización y movimientos del usuario.

Confidencialidad: No debe permitirse acceso a datos sensibles y personales del usuario.

Autenticación: No se debería permitir que usuarios no autorizados utilicen o accedan al sistema. Sólo las etiquetas NFC válidas de usuario deberían ser aceptadas por el lector.

Percepción y control del usuario: Proporcionar a los usuarios un control completo sobre sus datos privados. Permitirles decidir qué información debería ser incluida en sistemas y aplicaciones contratadas.

Facilidad de uso: Proporcionar a los usuarios una interfaz de funciones para gestionar fácilmente los datos personales. No saturar al usuario con programas complicados.

Flexibilidad y apertura: Brindar a los usuarios un control flexible para gestionar su información personal. Así mismo proporcionar la capacidad de elección de agentes o software desarrollado por compañas de gestión de privacidad de terceras partes.

2.4.2. Servicios Web

2.4.2.1. Introducción

En la actualidad existe una gran variedad de tecnologías para el desarrollo de aplicaciones Web, muchas de ellas incompatibles entre sí, por lo que pueden existir problemas a la hora de integrar aplicaciones de diferentes organizaciones que deban trabajar conjuntamente intercambiando información, entre los sistemas de información de una misma organización o incluso entre organizaciones y los sistemas de software utilizados por sus clientes.

Para poder llevar a cabo esta cooperación o intercambio de información, los desarrolladores de todas las partes deberían ponerse de acuerdo en cuanto a las tecnologías a utilizar. Esto haría muy compleja la cooperación y en ocasiones imposible si fueran más de dos las entidades que tuvieran que cooperar y necesitaran utilizar forzosamente diferentes arquitecturas de componentes y tecnologías para crear sus aplicaciones. Para solucionar estos problemas surgió el concepto de Servicio Web.

Los servicios Web según W3C son sistemas de software diseñados para permitir interoperabilidad máquina a máquina a través de una red. Estos servicios tienen interfaces descritas en un formato procesable por las máquinas que son accesibles a través de una red, como por ejemplo internet. Permiten a los clientes inter-operar con los servicios, los cuales se ejecutan en el sistema que los aloja.

Es decir, estos sistemas de software permiten la interoperabilidad y el intercambio de información entre aplicaciones, basándose en estándares abiertos (como XML) para la creación de los mensajes de comunicación y utilizando, protocolos de Internet (como HTTP, FTP o SMTP) para el transporte de los mismos. De esta forma se consigue que independientemente de la plataforma donde se ejecuten las aplicaciones (ya sea Windows, UNIX, Linux,…) y del lenguaje de programación concreto utilizado para crearlas (C++, Java, .NET,…), estas puedan comunicarse con otras aplicaciones a través de una red sin problemas.

Los Servicios Web están muy relacionados con el modelo de arquitectura orientada a servicios (SOA). SOA [22] [23] es un estilo de programación con el cual se pretende trasladar la idea de reutilización del código a la web para dar solución a los requisitos del negocio de forma distribuida. Se basa en el desarrollo del software mediante la programación de módulos auto contenidos y con funcionalidades muy reutilizables. Estos módulos interactúan con otros a través de la red para dar solución a los problemas a resolver. Por estas razones la mayoría de las veces suele identificarse a los Servicios Web como método de implementación en este tipo de arquitectura. No obstante, no es el único método, se pueden implementar usando cualquier otra tecnología basada en servicios (CORBA, DCOM, RPC).

Dentro de los Servicios Web existen dos estilos de diseño que se basan en este concepto pero que tienen diferencias significativas. Son los servicios web basados en SOAP y los Servicios Web basados en REST. A continuación se describirán los Servicios Web basados en REST ya que son los que se usarán en este proyecto.

2.4.2.2. Servicios Web basados en REST

Los servicios web basados en SOAP necesitan un protocolo propio de transporte para el intercambio de mensajes, pues con los servicios web basados en REST se pretende eliminar esa capa y utilizar únicamente el protocolo de transporte utilizado por la aplicación, por ejemplo HTTP.

La idea principal de los servicios REST es el uso de interfaz simple que utilice únicamente el conjunto estándar de operaciones (como son las del protocolo HTTP) para el intercambio de datos, siendo innecesario definir otro protocolo basado en patrones de intercambio de mensajes (como es el SOAP). Este tipo de servicio se centra en interactuar con recursos en vez de con mensajes y operaciones como sucedía con SOAP.

Un recurso es cualquier elemento de información en la red que represente un concepto de negocio y que posee un URI, es decir, un identificador uniforme de

recurso. Para manipularlos, clientes y servidores se comunican a través de una interfaz estándar (HTTP) e intercambian representaciones de los mismos. Una representación de de un recurso es una copia del estado público del recurso en un formato concreto. Los recursos pueden tener cero o más representaciones. Un ejemplo de recurso con su URI podría ser el siguiente:

http://www.librosweb.com/novelas/terror

Este recurso podría ser una colección de elementos que a su vez, pueden ser recursos con sus respectivas URIs y por ejemplo, dos representaciones, una HTML y otra JPEG.

http://www.librosweb.com/novelas/terror/eltrajedelmuerto

http://www.librosweb.com/novelas/terror/elresplandor

REST no es un estándar, es un estilo de arquitectura de software para sistemas hipermedias distribuidos tales como la Web que hace uso de estándares:

- HTTP
- URL
- XML, HTML, JPG, PNG, etc. para las representaciones de los recursos.
- Tipos MIME: text/xml, text/html, image/png, etc.

Un caso de arquitectura basada en REST que funciona a la perfección es la world wide web, o simplemente web. La motivación para crear servicios web basados en REST es la de emular a la propia web, utilizando sus principales características que hacen de la misma un éxito. A continuación se describirán los objetivos que se desean extraer de la web y las restricciones que se aplican en REST para alcanzarlos.

Objetivos y restricciones REST

Los principales objetivos de la web que se desean alcanzar en una arquitectura de software REST son:

- La escalabilidad en la interacción entre componentes (capacidad de un sistema para trabajar con diferentes cantidades de trabajo). La web ha crecido mucho desde su puesta en marcha y seguirá creciendo en el futuro. Todo ello sin empeorar su rendimiento incluso teniendo en cuenta la gran variedad de clientes que han aparecido con el tiempo: estaciones de trabajo, sistemas industriales, dispositivos móviles, etc.
- La generalidad de la interfaz. Gracias a que la web utiliza el protocolo de transporte HTTP cualquier cliente puede interactuar con cualquier servidor sin ninguna configuración especial, simplemente usando sus operaciones definidas.
- La extensibilidad del sistema (capacidad de un sistema para incluir mecanismos para la expansión/mejora del mismo con capacidades previstas sin tener que hacer cambios importantes en su infraestructura.) En la web está en constante crecimiento. Clientes y servidores están en funcionamiento durante años y es necesario que éstos sean capaces de interactuar con clientes y servidores más modernos y viceversa. La web lo consigue mediante el protocolo HTTP mediante el uso de cabeceras (cabeceras especiales como Accept o Content-Type pueden especificar que representaciones entienden el servidor y el cliente y que representación se usa en un mensaje concreto para transportar el estado del recurso.), a través de las URIs o mediante la habilidad para crear nuevos tipos de contenido.
- Compatibilidad con componentes intermedios. La web posee una infraestructura propia de componentes intermedios. La compatibilidad con componentes tales como cachés, firewalls y puertas de enlace (dispositivos que permite interconectar redes de computadoras con protocolos y arquitecturas diferentes a todos los niveles de comunicación.) ofrece una serie de ventajas como la mejora en rendimiento y seguridad y la posibilidad de encapsulamiento de sistemas.

Para poder conseguir los objetivos comentados, REST define un conjunto de restricciones o características que se deben seguir en el desarrollo de este tipo de servicios web:

- Uso de una interfaz uniforme. En REST los servicios no publican una serie de métodos u operaciones concretas para cada servicio como ocurría en los servicios basados en SOAP. Como se comentó anteriormente, REST se centra en recursos y para acceder a éstos existe una interfaz única y constante. Todos los recursos comparten las mismas operaciones. Las operaciones permiten manipular el estado público del recurso. En un sistema REST típico se define una interfaz con cuatro operaciones.
  - CREATE. Mediante esta operación un cliente manda al servidor una petición para crear un nuevo recurso. Opcionalmente el cliente puede mandar una representación del estado inicial de este recurso. El servidor responde con el URI del nuevo recurso.
  - DELETE. Con esta operación se envía una petición para eliminar un recurso del servidor. El usuario necesita saber el URI del recurso y tener permiso para hacerlo.
  - READ. Con esta operación el cliente envía una petición para obtener una representación del estado de un recurso, identificado con su URI. El cliente puede especificar qué tipos de representaciones entiende. Mediante esta operación se realiza una copia el estado del recurso en el servidor y se pega en el cliente. La copia del cliente no se mantiene sincronizada con el recurso en el servidor. El servidor puede cambiar el estado real del recurso y el cliente, de forma independiente, puede modificar su copia local del estado del recurso.
  - UPDATE. Como el servidor y el cliente tienen una copia diferente del estado, el cliente puede usar esta operación para sobrescribir o grabar su copia del estado en el servidor. De esta manera se puede actualizar el estado del recurso con las modificaciones hechas en el cliente.

Una de las características claves de los servicios web REST es el uso explícito de los métodos HTTP para ejecutar estas operaciones. Por ello se establece una asociación

uno-a-uno entre las operaciones create, delete, read y update y los métodos HTTP. De acuerdo a esta asociación:

- POST se asocia con la operación create.
- GET se asocia con la operación read.
- PUT se asocia con la operación update.
- DELETE se asocia con la operación delete.

- **Sin mantenimiento de estado.** Los servicios web basados en REST deben ser stateless que significa sin estado. Esto quiere decir que entre dos solicitudes cualesquiera entre un cliente y el servicio, este último no conserva ningún dato sobre el cliente. Para el servicio la segunda petición es una petición completamente nueva, no guarda sesiones de usuario. Es el propio usuario el que debe mantener el estado y enviarlo al servidor en cada solicitud. Esta forma de funcionar puede resultar tediosa pero tiene una gran ventaja, la escalabilidad. Para mantener el estado de un usuario hace falta memoria donde almacenarlo, en caso de que haya muchos usuarios el servidor podría quedarse sin memoria, por lo tanto podría llegar el momento en que no se pudieran admitir más clientes. Esto se podría solucionar utilizando varios servidores o bases de datos donde almacenar los estados pero son soluciones muy ineficientes que influyen mucho sobre el rendimiento, no recomendables si se desea tener un sistema altamente escalable.
- **Sintaxis universal para definir los recursos.** En un sistema basado en REST cada recurso debe ser únicamente direccionable a través de su URI. Como se comentó anteriormente, un URI es un identificador único de recurso que distingue un recurso de cualquier otro. Los URIs de los recursos de un servicio web basado en REST deben ser intuitivas y fáciles de adivinar de forma que un usuario sean capaz de entender a lo que apunta y los recursos derivados de éste. La forma de lograrlo es definir los URIs como una estructura al estilo de directorios, es decir una estructura jerárquica con un URI raíz y extendiendo sub-rutas de recursos.
- **Recursos con múltiples representaciones.** Cada recurso posee un estado interno que no es accesible por los usuarios del servicio. A lo que acceden los usuarios es a las diferentes representaciones posibles de dicho recurso. Una

representación es una copia del estado interno de un recurso en un formato concreto que se transfiere entre el usuario y el servicio. Es el desarrollador del servicio el que define qué tipos de representaciones son accesibles por los usuarios. La representaciones pueden ser en cualquier formato imaginable, tanto datos estructurados (documentos XML, HTML o JSON), como otro tipo de formatos tipo PNG, GIF, PDF, documentos Excel, etc. Las representaciones se definen mediante los atributos Accept y Content-Type de HTTP y tipos MIME. La mayoría de los tipos MIME son estándares, como XML o JSON. Como ya se vio en SOAP el usar protocolos y, en este caso tipos MIME estándar, facilita la interoperabilidad.

- **Componentes en capas:** Intermediarios, tales como firewalls, servidores caché, puertas de enlace, etc., pueden ser introducidos entre los clientes y los recursos, para ofrecer mejor rendimiento, seguridad, y mejorar la escalabilidad, ya que permiten mover funcionalidades de uso infrecuente hacia componentes intermedios mejorando el balanceo de carga.

¿Cómo funciona?

A continuación se mostrará un ejemplo del funcionamiento de un servicio web basado en REST. En el ejemplo, la empresa ficticia de venta de libros llamada "compratumusica" pone a disposición de sus clientes un servicio web REST que les permite obtener una lista de discos de música, obtener información detallada sobre un disco en particular y enviar una orden de compra.

Imaginemos que un cliente desea acceder a la lista de discos disponibles. En esta ocasión no existen operaciones del tipo getListadoDiscos() como puede ocurrir al crear un cliente a partir de un descriptor WSDL de un servicio SOAP. En este caso se accede mediante identificadores URI a los recursos disponibles directamente mediante la interfaz uniforme.

Para acceder al listado de discos se accede al recurso que será un listado de discos mediante su URI con el método de HTTP GET que como comentamos anteriormente

envía una petición para recibir una representación del recurso seleccionado. Quedaría de la siguiente manera:

GET https://www.compratumusica.com/discos

La generación del listado de libros disponibles es totalmente transparente al cliente. Todo lo que conoce es que ha solicitado el listado a través del URI y que el documento que contiene el mismo le ha sido devuelto. El desarrollador del servicio es libre de modificar la estructura de recursos que son accesibles sobre el servicio web sin que el cliente deba realizar ningún cambio. Seguirá utilizando la misma interfaz sencilla sobre la nueva estructura de recursos. De esta forma se consigue bajo acoplamiento entre las aplicaciones clientes y los servicios. El listado que sería devuelto sería como este:

```
<?xml version="1.0"?>
<l: Discos xmlns:l="https://www.compratumusica.com"
xmlns:xlink="http://www.w3.org/1999/xlink">
<Disco nombre="Don Quijote de la Mancha"
xlink:href="https://www.compratuslibros.com/libros/Don Quijote de la Mancha"/>
< Disco nombre="1984"
xlink:href=" https://www.compratuslibros.com/libros/1984"/>
< Disco nombre="2001 una odisea espacial"
xlink:href=" https://www.compratuslibros.com/libros/2001_una odisea espacial"/>
< Disco nombre="Juego de Tronos"
xlink:href=" https://www.compratuslibros.com/libros/Juego de  tronos"/>
</l: Discos >
```

Figura 7: RESPUESTA LISTADO DE LIBROS.

El servicio ha devuelto a la petición del cliente del recurso listado de libros una representación del mismo en formato XML. Como se observa cada libro en el listado posee el URI del recurso concreto, es decir de un libro específico. Como se comentó en un apartado anterior los URI de los recursos debes ser fáciles de seguir y adivinar. En este caso mediante el listado el cliente ya puede conocer el URI de los recursos que componen el listado, por lo que podrían hacer una petición sobre cualquiera de ellos fácilmente.

Ahora el usuario desea conocer la información de uno de los libros. Mediante el siguiente comando accede a su URI y recibe el resultado, una representación del libro elegido en formato XML.

```
GET https://www.compratuslibros.com/libros/1984
<?xml version="1.0"?>
<l:Libro xmlns:l="https://www.compratuslibros.com"
xmlns:xlink="http://www.w3.org/1999/xlink">
<ISBN>9788499890944</ISBN>
<Nombre>1984</Nombre>
<Autor>George Orwell</Autor>
<Editorial>Debolsillo</Editorial>
<Portada
xlink:href="https://www.compratuslibros.com/libros/1984/portada"/ >
<Lengua>Castellano</Lengua>
<Moneda>Euro</Moneda>
<Cantidad>7,55</Cantidad>
</l:Libro>
```

Figura 8: RESPUESTA DE UN SOLO LIBRO.

Con esta petición se devuelve la información de un libro en concreto. Como se puede observar sigue siendo sencillo encontrar los URIs. En este caso el usuario podría hacer la petición para obtener una representación de la portada del libro en cualquier otro formato de imágenes que estuviera definido en el servicio.

A continuación imaginemos que un usuario desea comprar el libro sobre el cual ha estado informándose. El usuario debe crear un documento de instancia de orden de compra ajustándose al esquema definido por la empresa y lo envía mediante la operación POST. El usuario tiene que enviar también su identificación porque como se explicó anteriormente el servicio no conserva el estado de la comunicación:

POST
https://www.compratuslibros.com/ordenesCompra/Orden.xml?id_usuario=1544

El servicio devolverá al usuario un URI donde el cliente puede encontrar la información de la orden en cualquier momento y así editarla o actualizarla mediante la operación PUT.

Por último la empresa desea eliminar un recurso del sistema porque va a dejar de vender uno de los libros. Usando la operación DELETE sobre el URI del recurso deseado se elimina del sistema.

DELETE https://www.compratuslibros.com/libros/1984

En el ejemplo comentado se accede a los recursos de forma directa mediante las operaciones de la interfaz simple de HTTP (GET, POST, PUT y DELETE), En un cliente programado en un lenguaje de alto nivel como Java podría se podrían utilizar librerías internas de Java como java.net.HttpURLConnection o javax.net.ssl.HttpsURLConnection para invocar las llamadas. Pero cuando se debe acceder a una gran cantidad de recursos esta manera puede resultar algo engorrosa. Por eso existen diferentes librerías que facilitan la integración de los clientes con las APIs REST de los proveedores. Algunas de ellas son Jersey, RESTEasy y Restlet.

A continuación se muestran las referencias a los contenidos que se han usado para explicar los servicios web [25] [26] [27].

2.5. Conclusión

Una vez visto el estado del arte y definidas las tecnologías que se van a utilizar en el desarrollo del proyecto, en el siguiente capítulo se va a describir el entorno de desarrollo, tanto la instalación como la configuración del mismo. Además se describirán el hardware y las versiones específicas de las herramientas y tecnologías utilizadas en el desarrollo y la ejecución del proyecto.

Sección 3

3. Entorno de desarrollo

3.1. Introducción

En esta sección se van a exponer las características de Hardware y Software que se han utilizado para el estudio, realización y ejecución de este proyecto. También contendrá un apartado de tutorial para la configuración del entorno para que funcione todo como se espera. Es importante seguir las indicaciones del tutorial para no tener problemas de compatibilidad y por lo tanto no poder ejecutar la aplicación desarrollada.

3.2. Hardware

Para el desarrollo y ejecución de la aplicación NFCInteractive se ha utilizado el siguiente Hardware:

- Portátil:
  - Fabricante: Toshiba
  - Modelo: Satellite PRO P50-B-V10
  - Procesador: i7
  - Memoria RAM: 8GB
  - Tarjeta Gráfica: Intel HD Graphics 520
- Smart Phone:
  - Fabricante: Samsung
  - Modelo: A3
  - Capacidad: 32gb
  - Memoria RAM: 2gb
  - NFC: Si
  - Bluetooth: Si
  - 4G: Si

- Wifi: Si
- Placa electrónica para simular funcionalidad:
- Fabricante: Arduino Mega 2560
- Etiquetas NFC:
- Modelo NTAG203
- Dimensión: Ø25mm
- Material: blanco Polypropylen - adhesivo
- Chip: NXP NTAG203 - NFC Forum Type 2 Tag - ISO 14 443-2 A, ISO 14 443-3 A
- Capacidad de memoria: 168 Byte - NDEF formateada: 137 Byte
-
- Hardware adicional:
- Ratón Bluetooth Microsoft
- Rooter Belkin para conexión Smart Phone-Servidor

3.3. Software

En este apartado se hablará de las aplicaciones, bibliotecas, Plugins, Software de soporte ofimático, etc. que se ha usado para el desarrollo de este proyecto. A menudo, cuando se habla de Software, sólo se piensa en la aplicación que se está desarrollando o el entorno que se usa para desarrollar, sin embargo, como veremos a continuación es necesario apoyarse en gran cantidad de Software de terceros para llevar a cabo un desarrollo completo y de buena calidad.

3.3.1. Sistema operativo

El sistema operativo es el software fundamental para llevar a cabo este proyecto, ya que sin él no se podría ejecutar ninguna aplicación y por lo tanto es lo primero que se necesita. Dado que se trata de una aplicación servidor-cliente y el cliente es un dispositivo móvil, serán necesarios dos sistemas operativos, uno de ellos será el sistema que albergará la máquina que se usará para diseñar las aplicaciones necesarias (se usará el mimo equipo para desplegar la aplicación del servidor) y el otro específico para el dispositivo móvil.

El sistema operativo que se usará en el equipo de desarrollo y despliegue será Windows 7 y el sistema operativo móvil será Android Lollipop.

3.3.2. IDE

Un IDE (Integrated Development Environment) es una herramienta fundamental para el desarrollo de aplicaciones de todo tipo, pues sin este tipo de Software sería muy complejo el hecho de desarrollar aplicaciones. Los IDE tienen como objetivo principal facilitar la tarea de desarrollo delegando algunas tareas en el propio IDE, además, un buen IDE muestra los errores que el programador comete sobre la marcha para que le sea más fácil de detectar, comprueba que todos las referencias a librería se encuentren disponibles y accesibles, ayuda a estructurar el código de una forma más estándar para ayudar a detectar fácilmente zonas del código en caso de buscar en el mismo, entre otras.

En este caso en concreto, se necesita un IDE para desarrollar una aplicación móvil Android, una aplicación web desarrollada en Java y una pequeña aplicación desarrollada para nuestro dispositivo Arduino con el fin de simular alguna funcionalidad de la aplicación. Para esta combinación, se ha optado por los siguientes IDEs:

- **Eclipse Versión Luna Service Release 2 (4.4.2)**

Uno de los IDE más potentes que existen de distribución libre y gratuita. Soporta prácticamente todos los lenguajes de programación alto y bajo nivel. Tiene un apoyo enorme por parte de una gran comunidad y por lo tanto prácticamente cualquier duda está resuelta en multitud de foros. Se añadirá el Plugin de Android para desarrollar también la aplicación Android que correrá en el móvil, durante el desarrollo de la misma, se podrá optar por ejecutar la aplicación en los emuladores que proporciona el Plugin de Android o directamente en un dispositivo móvil.

- **Arduino IDE 1.6.5**

Entorno ofrecido por Arduino para el desarrollo de código para los dispositivos Arduino, este entorno es solo y exclusivamente para Arduino, lleva incorporado los drivers para los distintos dispositivos Arduino.

3.3.3. JDK

Ya que las plataformas usadas para el desarrollo son Java y Android, será necesario instalar la máquina virtual de Java para desarrolladores, esta máquina se llama JDK (Java Development Kit) que a diferencia de la JRE (Java Runtime Enviroment), la JDK está más enfocada a los usuarios desarrolladores mientras que la JRE es para los usuarios consumidores de software. En caso de este proyecto, se instalará la última versión disponible (V. 8.45u).

3.3.4. Android SDK

Conjunto de herramientas de desarrollo proporcionado por Google para desarrollar aplicaciones Android, este kit de herramientas es la base para crear una aplicación Android, no sólo eso sino que sin este kit no es posible llevar a cabo esta tarea. Android SDK proporciona todas las APIs disponibles para interactuar con los dispositivos Android, contiene un simulador que nos permitirá ejecutar nuestra aplicación de una forma virtual, documentación acerca de cada una de las APIs proporcionadas, etcétera.

La versión usada en este proyecto es la última disponible en la página web de google (V 24.3.3) [https://developer.android.com/sdk/index.html#Other], esta elección se debe a que como ya se ha comentado anteriormente, la aplicación va a estar disponible también para la última versión de Android (Lollipop), las APIs para esta versión de Android están contenidas en esta versión de SDK.

3.3.5. Servidores

Para la realización de la tarea propuesta en este proyecto se ha visto necesario el uso de dos servidores distintos, uno de ellos es para ejecutar la aplicación web que se encargará por un lado de gestionar los fichajes solicitados por el usuario y por otro también gestionará la parte de abrir puertas o lo que se ha denominado como llave personal, el otro servidor es necesario para la gestión de base de datos, es decir, almacenar los datos de fichaje así como la información necesaria para realizar los accesos.

Como servidor de aplicaciones se utilizará GlassFish Server y el sistema gestor base de datos se utilizará MySQL Server. A continuación se realizará una breve descripción de cada uno de ellos.

- **GlassFish Server**

Es un servidor de aplicaciones de software libre desarrollado por Sun Microsystems, adquirida por Oracle Corporation, que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación. Es gratuito, de código abierto y de libre distribución bajo licencias CDDL y GNU GPL. También existe una versión comercial denominada Oracle GlassFish Enterprise Server (antes Sun GlassFish Enterprise Server).

Se usará la versión V 4.0 para ejecutar la aplicación objeto de este proyecto, como ya se ha comentado anteriormente, la aplicación que correrá en este servidor constará de una parte que será el modelo de datos en donde se definirán los tipos de datos, es decir, clases y objetos DAO que se usarán para realizar distintas gestiones con la base de datos y albergará también los servicios web basados en REST, esta última parte será invocada por la aplicación móvil y será el punto clave para la conexión entre el servidor y la aplicación móvil.

- **MySQL Server**

A la hora de tener la necesidad de hacer persistentes los datos que maneja una aplicación se puede optar por varias soluciones, entre ellas se pueden utilizar ficheros, un problema típico de este planteamiento es la gestión de estos datos a medida que van creciendo. Como alternativa a la primera propuesta sería el uso de un sistema gestor de base de datos, estos sistemas permiten gestionar gran cantidad de información de una forma sencilla y eficiente.

Estudiado el problema concreto de este proyecto, se ha visto más adecuado el uso de un sistema gestor de base de datos relacional, esta decisión se debe a varias razones, entre ellas la escalabilidad y la flexibilidad que estos sistemas ofrecen. La información que interesa almacenar en este proyecto es fundamental para el funcionamiento de la aplicación. Para poder llevar a cabo el control de presencia o el control de acceso es necesario comprobar que se tienen permisos, para ello se verificarán dichos permisos en la base de datos.

Se usará como servidor de base de datos MySQL Server en su versión 5.6, este sistema de gestión de bases de datos es relacional, multihilo y multiusuario. Es un sistema desarrollado por Sun Microsystems y que actualmente ha adquirido Oracle Corporation, es gratuito, de código abierto y de libre distribución bajo licencia GNU GPL aunque también existe una licencia de uso para empresas que deseen soporte.

3.3.6. Bibliotecas

Se ha comentado en más de una ocasión la arquitectura que se va a utilizar para este proyecto, para realizar distintas conexiones por ejemplo con la base de datos o con el dispositivo de Arduino, hace falta usar una serie de librerías que gracias a las cuales harán posible estas conexiones.

Las bibliotecas simplifican mucho el trabajo del desarrollador ya que parte del código que hay que añadir para conseguir cierta funcionalidad está integrada en dicha biblioteca y simplemente hay que realizar las distintas llamadas a las funciones contenidas en ellas para conseguir esa funcionalidad.

En este proyecto se van a usar dos librerías fundamentales:

- **JDBC**

JDBC (Java DataBase Connectivity) es una librería Java que permite, como su propio nombre indica, realizar operaciones sobre bases de datos mediante el lenguaje Java. Entre las operaciones más generales se encuentran establecer la conexión a la base de datos, enviar sentencias SQL y procesar los resultados. Para utilizar esta API únicamente es necesario importarla dentro del código donde se vaya a utilizar, no es necesario descargarla ya que viene incluida en el JDK. No obstante hay que tener en cuenta que la biblioteca JDBC hace uso del driver que proporciona MySQL, se explicará más adelante el procedimiento para incluirlo dentro del proyecto para realizar estas operaciones.

- **RXTX**

Es una librería diseñada para comunicar dispositivos por un canal de transmisión en serie. Provee distintas funciones; conexión con entre los dispositivos, envío y recepción de datos. Para poder realizar comunicación con Arduino, es necesario esta librería u otra similar con el fin de transmitir información entre el servidor y Arduino cuando se dé el caso. El servidor, una vez comprobadas las credenciales, envía una señal al dispositivo Arduino para que éste encienda una luz (simulación de la apertura de una puerta o caja fuerte, etc.), para hacer posible esta operación es necesario comunicar el servidor con Arduino. Cabe destacar que el dispositivo de Arduino se conectará físicamente al servidor vía USB (Universal Serial Bus).

Para poder usar esta biblioteca es necesario agregar los drivers que proporcionan los desarrolladores en su página web [http://rxtx.qbang.org/wiki/index.php/Download]. Más adelante se explicará de una forma más extendida la forma de instalar los drivers y agregar la biblioteca al proyecto para su uso.

Hay muchas bibliotecas para la gestión de las comunicaciones en serie, la usada en este proyecto es una biblioteca creada por una comunidad de desarrolladores y es de distribución libre y gratuita bajo la licencia LGPL. Toda la documentación y ejemplos de uso se encuentran en la página oficial.

3.3.7. Software adicional

- **MySQL WorkBench 6.3**

Esta aplicación es de gran ayuda para interactuar con la base de datos, tiene una interfaz gráfica amigable y presenta gran cantidad de herramientas para la gestión, interacción y mantenimiento de la base de datos. MySQL trae consigo una herramienta muy sencilla para su gestión, tan sencilla es la herramienta que trae que se ejecuta sobre la consola de Windows.

Esta herramienta o conjunto de herramientas lo proporciona Oracle en su página oficial [https://www.mysql.com/products/workbench/], es gratuita y se distribuye bajo licencia GPL.

- **Microsoft office 2010**

Software ofimático de Microsoft, se ha utilizado para la redacción de la presente memoria.

- **Microsoft Visio**

Software de Microsoft para realizar distintos diagramas. Se ha usado para realizar los diagramas que se adjuntan a esta memoria,

- **Microsoft Project**

Para crear un buen plan de ejecución de las distintas tareas de un proyecto nunca es una tarea fácil, sin embargo esta aplicación hace que sea mucho más sencillo realizar esta tarea mostrando las tareas, quién las ejecuta y el tiempo de duración de cada una de ellas. Trae consigo multitud de funcionalidades para controlar que un proyecto se ejecute según lo estimado, de lo contrario avisará de la falta de recursos. Se ha usado Microsoft Project para realizar el diagrama de Gantt adjunto en esta memoria.

3.4. Conclusión

En esta sección se ha visto en detalle el entorno de desarrollo así como los distintos elementos que lo conforman y la configuración del mismo para la correcta ejecución del proyecto. Con la presente sección se pretende sirva como referencia para la instalación de este tipo de entornos.

Todo el software utilizado para el desarrollo de este proyecto es gratuito a excepción de las herramientas de Microsoft que son de pago, dos de ellas (Visio y Project) se han obtenido por medio del portal de la universidad ya que hay un convenio con la empresa para proporcionar este software de forma gratuita a los estudiantes, mientras que el Office se ha obtenido adquiriendo la licencia de uso personal junto con el equipo.

Sección 4

4. Análisis del sistema

4.1. Introducción

En la primera fase del proyecto se planteó el problema y se buscaron posibles soluciones, además, se incluyó información acerca del estudio y aprendizaje de diferentes tecnologías disponibles para resolver dicho problema.

En esta sección se recoge la información correspondiente a la segunda fase del proyecto, la fase de análisis. En esta fase, una vez que se tiene una idea más centrada de lo que se va a desarrollar y familiarizado con las herramientas y tecnologías a utilizar, se realizará la consecución de casos de uso y requisitos de software.

Esta sección es un resumen de los documentos de casos de uso y requisitos de software que se obtuvieron durante la realización de la misma. Para ver los documentos completos habrá que dirigirse a los apéndices, al final de esta memoria, donde están incluidos dichos documentos de forma completa.

4.2. Requisitos del software.

Los requisitos de software recogerán todos los aspectos relevantes y aquellos definidos por el usuario que debe cumplir la aplicación. Sirven como guía de referencia al desarrollador y le permitirán validar las funciones desarrolladas comprobando de esta forma la obtención del producto deseado por el cliente.

Se ha propuesto definir dos grupos de requisitos; requisitos funcionales y no funcionales. Los requisitos funcionales son aquellos que definen las funcionalidades del sistema a desarrollar y los no funcionales definen otros aspectos del sistema como usabilidad, seguridad, etcétera.

Cada requisito es identificado por un código único, un título, una descripción del mismo, el origen del requisito, la necesidad y la prioridad que tiene.

El identificador consta de un código alfabético RF o RNF que indica si es requisito funcional o no funcional respectivamente, seguido de un guion y un número único dentro del tipo de requisito.

El título representa una pequeña descripción del requisito.

El origen con este atributo se pretende indicar el origen del requisito que puede ser cliente, programador o analista.

Con el atributo **Necesidad** se pretende resaltar la necesidad del requisito dentro del proyecto y puede tomar tres posibles valores: **Necesario**, que indica que es muy importante, es decir, imprescindible para alcanzar el objetivo; **Deseable**, que corresponderá a los requisitos que son deseables pero no imprescindibles para el objetivo del proyecto y **Opcional**, para los requisitos que no influirían en el objetivo del proyecto si desaparecieran.

Prioridad, este atributo indica el grado de importancia y estable por tanto qué requisitos son más importantes. Los posibles valores que puede tomar este atributo son: alta, media o baja. Aquellos que se establezcan como prioritarios o prioridad alta serán ejecutados antes que el resto y los que se identifiquen como baja serán los últimos en ejecutarse.

Descripción: Explicación textual del requisito que será clara y concisa para que al leerla se pueda comprender el requisito fácil y rápidamente.

Identificador de requisito	**Título**	
Origen:	**Necesidad:**	**Prioridad:**
Descripción		

Esquema requisito

A continuación, se expondrán los requisitos ordenados por un identificador único para cada uno de ellos. En el documento de requisitos se expondrán los requisitos de software en formato extendido y agrupados por funcionalidad.

4.2.1. Requisitos funcionales

RF-0001	Definir modos	
Origen: Usuario	**Necesidad: Necesario**	**Prioridad: Alta**
Usuario deberá poder definir modos predeterminados y guardarlos en una etiqueta NFC.		

Tabla 3. RF-0001-Definir modos

RF-0002	Aplicar modos	
Origen: Usuario	**Necesidad: Necesario**	**Prioridad: Alta**
El usuario deberá poder aplicar un modo recuperándolo desde una etiqueta NFC.		

Tabla 4. RF-0002-Aplicar modos

RF-0003	Modos predeterminados	
Origen: Usuario	**Necesidad: Necesario**	**Prioridad: Alta**
El usuario podrá escoger de la lista de modos predeterminados el que desee y se definirán en el requisito RF-0005 , RF-0006 , RF-0007 , RF-0008 y RF-0009		

Tabla 5. RF-0003-Modos predeterminados

RF-0004	Modo definido por el usuario	
Origen: Usuario	**Necesidad: Deseable**	**Prioridad: Media**

El usuario podrá definir un modo seleccionando entre los siguientes elementos:
• Estado del teléfono:
  ○ Normal
  ○ Silencio
  ○ Solo vibración
• Estado de la Wifi
  ○ Habilitado o no		

- Estado del Bluetooth
  - Habilitado o no
- Estado de datos móviles
  - Habilitado o no
- Estado del GPS
  - Habilitado o no

Tabla 6. RF-0004-Modo definido por el usuario

RF-0005	Modo Normal	
Origen: Usuario	**Necesidad: Necesario**	**Prioridad: Alta**
Este modo define que los avisos -sean llamadas o notificaciones- con sonido y vibración, habilita Wifi, deshabilita Bluetooth, deshabilita GPS y deshabilita Datos móviles		

Tabla 7. RF-0005-Modo normal

RF-0006	Modo Avión	
Origen: Usuario	**Necesidad: Necesario**	**Prioridad: Alta**
Este modo activa el modo avión en el dispositivo móvil. Deshabilita todas las comunicaciones.		

Tabla 8. RF-0006-Modo avión

RF-0007	Modo Reunión	
Origen: Usuario	**Necesidad: Necesario**	**Prioridad: Alta**
Este modo silencia los avisos de llamada y las notificaciones, deshabilita la vibración del Smart Phone y GPS. Mantiene el estado por defecto de Wifi, Bluetooth y Datos móviles.		

Tabla 9. RF-0007-Modo reunión

RF-0008	Modo Coche	
Origen: Usuario	**Necesidad: Necesario**	**Prioridad: Alta**

El modo Coche activa el sonido para las notificaciones. Deshabilita Wifi y habilita GPS, Bluetooth y datos móviles.

Tabla 10. RF-0008-Modo coche

RF-0009	Modo Casa	
Origen: Usuario	**Necesidad: Necesario**	**Prioridad: Alta**
El modo Casa activa el sonido para las notificaciones. Habilita Wifi y Bluetooth. Deshabilita GPS y datos móviles.		

Tabla 11. RF-0009-Modo casa

RF-0010	Usuario estándar	
Origen: Usuario	**Necesidad: Necesario**	**Prioridad: Alta**
Todos los usuarios tendrán acceso a la opción de modos de la aplicación. Podrán definir los modos, guardarlos y aplicarlos.		

Tabla 12. RF-0010-Modo estándar

RF-0011	Control de presencia en base a una etiqueta NFC	
Origen: Usuario	**Necesidad: Necesario**	**Prioridad: Alta**
La aplicación deberá llevar a cabo un control de presencia en función de una etiqueta que contendrá la información del centro o lugar que se desee registrar sin que el usuario introduzca ningún dato adicional.		

Tabla 13. RF-0011-Control de presencia en base a una etiqueta NFC

RF-0012	Control de presencia: registro de fecha, hora y lugar	
Origen: Usuario	**Necesidad: Necesario**	**Prioridad: Alta**
Cuando un usuario solicite realizar fichaje, el sistema registrará la fecha y la hora con una precisión de segundos y la ubicación GPS del lugar desde el cual se hace el fichaje		

Tabla 14. RF-0012-Control de presencia: registro de fecha, hora y lugar

RF-0013	Control de presencia: usuarios registrados	
Origen: Usuario	**Necesidad: Necesario**	**Prioridad: Alta**

Para que un usuario pueda tener acceso a la opción de control de presencia, éste deberá ser registrado y validado previamente por un administrador.

Tabla 15. RF-0013-Control de presencia: usuarios registrados

RF-0014	Control de presencia: usuarios registrados asignados a empresas	
Origen: Usuario	**Necesidad: Necesario**	**Prioridad: Alta**
El usuario podrá llevar a cabo el control de presencia siempre y cuando esté asociado con al menos una empresa.		

Tabla 16. RF-0014-Control de presencia: usuarios registrados asignados a empresas

RF-0015	Permisos sobre usuarios y objetos	
Origen: Usuario	**Necesidad: Necesario**	**Prioridad: Alta**
El administrador del sistema deberá definir permisos de acceso sobre los usuarios y objetos registrados estableciendo qué usuarios tienen permisos sobre qué objetos.		

Tabla 17. RF-0016-Permisos sobre usuarios y objetos

RF-0016	Control de acceso: Usuarios	
Origen: Usuario	**Necesidad: Necesario**	**Prioridad: Alta**
La opción de Control de acceso estará disponible únicamente para aquellos usuarios que hayan sido registrados y validados por un administrador.		

Tabla 18. RF-0017-Control de acceso: Usuarios

RF-0017	Permitir acceso	
Origen: Usuario	**Necesidad: Necesario**	**Prioridad: Alta**
El sistema deberá permitir acceso a los usuarios registrados sobre aquellos objetos solicitados cuando tengan permiso definido en el sistema.		

Tabla 19. RF-0018-Permitir acceso

RF-0018	Acceso desde una etiqueta NFC	
Origen: Usuario	**Necesidad: Necesario**	**Prioridad: Alta**

La aplicación deberá llevar a cabo un control de acceso en función de una etiqueta que contendrá la información del objeto al que se desee acceder sin que el usuario introduzca ningún dato adicional.

Tabla 20. RF-0020-Acceso desde una etiqueta NFC

4.2.2. Requisitos no funcionales

4.2.2.1. Requisitos del sistema

RNF-0101	Sistema operativo del dispositivo móvil	
Origen: Analista	**Necesidad: Necesario**	**Prioridad: Alta**
La aplicación móvil se desarrollará para el sistema operativo Android y para versiones v5.0 – Lollipop y posteriores.		

Tabla 21. RNF-0101-Sistema operativo del dispositivo móvil

RNF-0102	Aplicación servidor	
Origen: Analista	**Necesidad: Deseable**	**Prioridad: Alta**
El sistema encargado de almacenar y verificar los datos se alojará en un servidor centralizado y se accederá a él mediante servicios web basados en REST.		

Tabla 22. RNF-0102-Aplicación servidor

RNF-0103	Servidor de base de datos	
Origen: Analista	**Necesidad: Deseable**	**Prioridad: Media**
El sistema usará el SGBD MySQL server V 5.6 como servidor de base de datos.		

Tabla 23. RNF-0103-Servidor de base de datos

RNF-0104	Servidor de aplicaciones	
Origen: Analista	**Necesidad: Deseable**	**Prioridad: Media**
El sistema usará como servidor de aplicaciones GlassFish Server V 4.0		

Tabla 24. RNF-0104-Servidor de aplicaciones

RNF-0105	IDE	

Origen: Programador	Necesidad: Opcional	Prioridad: Baja
El sistema se desarrollará usando el IDE Eclipse v 4.4.2 y Arduino 1.6.5.		

Tabla 25. RNF-0105-IDE

RNF-0106	NFC	
Origen: Analista	**Necesidad: Necesario**	**Prioridad: Media**
La aplicación móvil sólo funcionará si está habilitado NFC en el dispositivo móvil.		

Tabla 26. RNF-0106-NFC

4.2.2.2. Requisitos de Hardware

RNF-0201	Dispositivo móvil	
Origen: Analista	**Necesidad: Necesario**	**Prioridad: Alta**
Es necesario el uso de un dispositivo móvil compatible con la tecnología NFC y sistema operativo Android.		

Tabla 27. RNF-0201-Dispositivo móvil

RNF-0202	Etiquetas NFC	
Origen: Analista	**Necesidad: Necesario**	**Prioridad: Media**
Es necesario el uso de etiquetas NFC que permitan lectura y escritura para almacenar información en ellas y consultarlas cuando se precise.		

Tabla 28. RNF-0202-Etiquetas NFC

RNF-0203	Placa Arduino	
Origen: Usuario	**Necesidad: Deseable**	**Prioridad: Media**
Para realizar la simulación del control de acceso será necesario usar una placa Arduino Mega 2560 o similar.		

Tabla 29. RNF-0203-Placa Arduino

RNF-0204	Equipo servidor	
Origen: Analista	**Necesidad: Deseable**	**Prioridad: Media**

El servidor se ejecutará sobre una máquina con las siguientes características: • Sistema operativo Windows 10 o 2008R Server. 64bits. • Memoria RAM: min 1GB. • Procesador: mínimo 2 núcleos.

Tabla 30RNF-0204- Equipo servidor

4.2.2.3. Requisitos de usabilidad

RNF-0301	Aviso resultado operaciones	
Origen: Usuario	**Necesidad: Necesario**	**Prioridad: Media**
Mantener informado al usuario del estado de toda operación que realice con la aplicación móvil.		

Tabla 31. RNF-0301-Aviso resultado operaciones

RNF-0302	Ayuda al leer etiquetas NFC	
Origen: Usuario	**Necesidad: Necesario**	**Prioridad: Media**
El sistema deberá indicar el momento en que el usuario debe acercar su dispositivo a una etiqueta NFC para leer su contenido o grabar información en ella.		

Tabla 32. RNF-0302-Ayuda al leer etiquetas NFC

RNF-0303	Comprobar estado de conexión con el servidor	
Origen: Analista	**Necesidad: Necesario**	**Prioridad: Media**
El sistema comprobará el estado de la conexión con el servidor antes de realizar una operación en el mismo.		

Tabla 33. RNF-0303-Comprobar estado de conexión con el servidor

RNF-0304	Identificación fácil de los elementos que se pulsen	
Origen: Usuario	**Necesidad: Opcional**	**Prioridad: Baja**
Al pulsar un elemento del menú, éste cambiará el color de su fondo. Al pulsar un botón bien cambiará su color de fondo, su tamaño o las dos cosas a la vez.		

Tabla 34. RNF-0304-Identificación fácil de los elementos que se pulsen

RNF-0305	Icono volver atrás	
Origen: Usuario	**Necesidad: Opcional**	**Prioridad: Media**
La aplicación móvil mostrará un icono en la parte superior derecha para volver atrás en todas las pantallas excepto en la principal.		

Tabla 35. RNF-0305-Icono volver atrás

RNF-0306	Aviso de NFC deshabilitado	
Origen: Analista	**Necesidad: Deseable**	**Prioridad: Media**
El sistema mostrará una ventana modal cuando se desactive NFC con un mensaje indicando que NFC está deshabilitado.		

Tabla 36. RNF-0306-Aviso de NFC deshabilitado

4.3.Casos de uso

4.3.1. Casos de uso en formato simple

En esta sección se expondrán los distintos actores, casos de uso resumidos y los diagramas de casos de uso que muestran la interacción entre los actores y el sistema.

A continuación se muestran los actores que participan en el sistema:

- Usuario estándar: Es el actor potencial del sistema, interactuará con la aplicación móvil mediante los distintos menús para el uso del módulo de cambios de estado del teléfono, podrá crear modos, guardarlos en las etiquetas NFC y sobre todo aplicar los modos definidos en el móvil. Sobre este usuario no se tiene ningún dato, es decir, no necesita registro alguno.
- Usuario registrado: Es el actor que interactúa con la aplicación móvil y en función de los permisos que tenga podrá hacer uso -además del módulo de cambios de estado- de los módulos de control de presencia y de control de acceso.

- Administrador: Este es el actor encargado de administrar los datos de las empresas, usuarios y objetos que se pretenda tener acceso a ellos en la base de datos. Sin esta tarea, el sistema no funcionaría correctamente.

Se han obtenido los siguientes casos de uso:

- Definir modo: Recoge el escenario de elegir un modo definido o definir uno y grabarlo en una etiqueta NFC.
- Aplicar modo: Este escenario muestra las acciones llevadas a cabo para que a partir de la información contenida en una etiqueta el usuario pueda cambiar el estado de su teléfono. El modo aplicado dependerá de la información contenida en la etiqueta.
- Fichar (Control de presencia): Define el escenario de realizar el marcaje o fichar la entra o salida de un puesto de trabajo basándose en la información contenida en la etiqueta NFC y en el código de enlace del dispositivo móvil con la cuenta de usuario.
- Acceso (Control de acceso): Muestra el escenario de dar una señal de apertura a un mecanismo de cierre de puertas, caja fuerte, etc. desde la aplicación simplemente basándose en la información contenida en una etiqueta NFC y el código de enlace del dispositivo móvil con la cuenta de usuario.
- Administración del sistema: Representa el escenario en el que un usuario administra el sistema dando de alta usuarios, empresas, enlaza dispositivos con las cuentas de los usuarios etcétera.

A continuación se muestra el diagrama de los casos de uso descritos anteriormente:

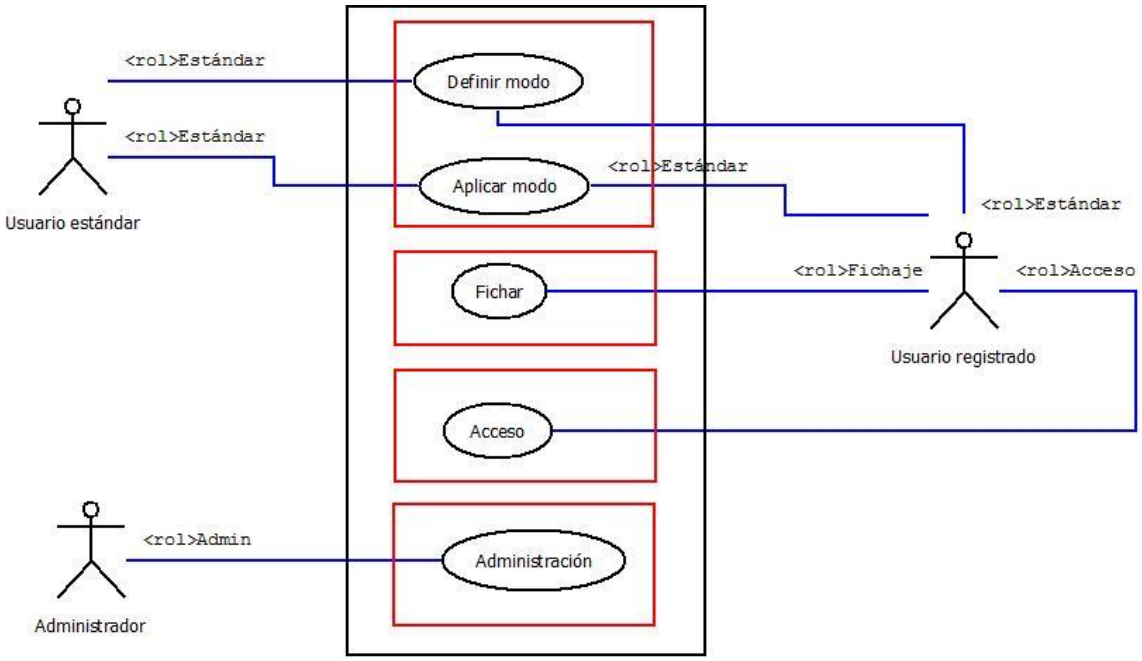


Figura 9: DIAGRAMA DE CASOS DE USO

4.3.2. Casos de uso en formato extendido

En este apartado se expondrán los casos de uso en forma extendido con el fin de detallar cada caso lo máximo posible.

CASO DE USO	Guardar modo predefinido en una etiqueta NFC		
IDENTIFICADOR	CU-001		
ACTORES	Usuario estándar, Usuario registrado		
PRE-CONDICIÓN	-		
ESCENARIO DE ÉXITO		ACTORES	SISTEMA
	1	El usuario seleccionará la opción de “Definir modos”	
	2	El usuario seleccionará la opción “Normal”	
	3		El sistema mostrará ayuda para proceder al guardado de la información.

	4	El usuario acercará el dispositivo móvil a la etiqueta NFC.	
	5		El sistema guardará la información en la etiqueta NFC y mostrará un mensaje informando de ello.
ESCENARIO ALTERNATIVO		ACTORES	SISTEMA
	1	El usuario seleccionará la opción de “Definir modos”	
	2	El usuario seleccionará la opción “Normal”	
	3		El sistema mostrará ayuda para proceder al guardado de la información.
	4	El usuario acercará el dispositivo móvil a la etiqueta NFC.	
	5		El sistema detecta algún error al intentar guardar la información en la etiqueta NFC y mostrará un mensaje informando de ello.
POST-CONDICIÓN	El modo seleccionado por el usuario quedará guardado en la etiqueta NFC.		

Tabla 37. CU-001 - Guardar modo predefinido en una etiqueta NFC

CASO DE USO	Guardar modo nuevo en una etiqueta NFC
IDENTIFICADOR	CU-002

ACTORES	Usuario estándar, Usuario registrado		
PRE-CONDICIÓN	-		
ESCENARIO DE ÉXITO		ACTORES	SISTEMA
	1	El usuario seleccionará la opción de “Definir modos”	
	2	El usuario seleccionará la opción “Definir nuevo”	
	3		El sistema mostrará las opciones para confeccionar un modo.
	4	El usuario seleccionará las siguientes opciones: Modo de audio = silencio, Wifi = si, Bluetooth = si, Datos = si y GPS = si	
	5	El usuario pulsará la opción de “Guardar”	
	6		El sistema mostrará ayuda para proceder al guardado de la información.
	7	El usuario acercará el dispositivo móvil a la etiqueta NFC.	
	8		El sistema guardará la información en la etiqueta NFC y mostrará un mensaje informando de ello.
ESCENARIO ALTERNATIVO		ACTORES	SISTEMA

	1	El usuario seleccionará la opción “Definir modos”	
	2	El usuario seleccionará la opción “Definir nuevo”	
	3		El sistema mostrará las opciones para confeccionar un modo.
	4	El usuario seleccionará las siguientes opciones: Modo de audio = silencio, Wifi = si, Bluetooth = si, Datos = si y GPS = si	
	5	El usuario pulsará la opción de “Guardar”	
	6		El sistema mostrará ayuda para proceder al guardado de la información.
	7	El usuario acercará el dispositivo móvil a la etiqueta NFC.	
	8		El sistema detecta algún error al intentar guardar la información en la etiqueta NFC y mostrará un mensaje informando de ello.
POST-CONDICIÓN	El nuevo modo confeccionado por el usuario quedará guardado en la etiqueta NFC.		

Tabla 38. CU-002 - Guardar modo nuevo en una etiqueta NFC

CASO DE USO	Aplicar un modo desde una etiqueta NFC
IDENTIFICADOR	CU-003

ACTORES	Usuario estándar, Usuario registrado		
PRE-CONDICIÓN	Etiqueta NFC con información de modo guardada		
ESCENARIO DE ÉXITO		ACTORES	SISTEMA
	1	El usuario desbloqueará la pantalla del dispositivo móvil.	
	2	El usuario acercará el dispositivo móvil a la etiqueta NFC	
	3		El sistema leerá la etiqueta y aplicará el modo notificando al usuario dicho cambio.
ESCENARIO ALTERNATIVO		ACTORES	SISTEMA
	1	El usuario desbloqueará la pantalla del dispositivo móvil.	
	2	El usuario acercará el dispositivo móvil a la etiqueta NFC	
	3		El sistema leerá la etiqueta y al comprobar que la información contenida está corrupta o no se corresponde con un modo notificará error de lectura.
POST-CONDICIÓN	El modo contenido en la etiqueta NFC se aplicará al dispositivo móvil.		

Tabla 39. CU-003 - Aplicar un modo desde una etiqueta NFC

CASO DE USO	Control de presencia
IDENTIFICADOR	CU-004

ACTORES	Usuario registrado		
PRE-CONDICIÓN	Etiqueta NFC con código de empresa y usuario perteneciente a dicha empresa.		
ESCENARIO DE ÉXITO		ACTORES	SISTEMA
	1	El usuario seleccionará la opción “Control de presencia”	
	2		El sistema comprobará si existe conexión con el servidor.
	3		El sistema mostrará ayuda para proceder a la lectura de la información contenida en la etiqueta NFC.
	4	El usuario acercará el dispositivo móvil a la etiqueta NFC	
	5		El sistema leerá la etiqueta, junto con el token de usuario y las coordenadas GPS los enviará al servidor donde una vez comprobados los datos, guardará fecha, hora y coordenadas GPS en la base de datos.
	6		El sistema mostrará un mensaje informando al usuario de que el control se ha llevado a cabo de forma correcta.
		ACTORES	SISTEMA

ESCENARIO ALTERNATIVO	1	El usuario seleccionará la opción “Control de presencia”	
	2	El usuario acercará el dispositivo móvil a la etiqueta NFC	
	3		El sistema comprobará si existe conexión con el servidor.
			El sistema mostrará un mensaje de error de conexión con el servidor.
POST-CONDICIÓN	Fecha, hora y coordenadas GPS junto con el usuario guardados en la correspondiente tabla en la base de datos.		

Tabla 40 .CU-004 - Control de presencia

CASO DE USO	Control de acceso		
IDENTIFICADOR	CU-005		
ACTORES	Usuario registrado		
PRE-CONDICIÓN	Etiqueta NFC con código de objeto al que se quiere acceder y usuario con premiso definido para poder acceder.		
ESCENARIO DE ÉXITO		ACTORES	SISTEMA
	1	El usuario seleccionará la opción “Control de acceso”	
	2		El sistema comprobará si existe conexión con el servidor.
	3		El sistema mostrará ayuda para proceder a la lectura de la información contenida en la etiqueta NFC.

	4	El usuario acercará el dispositivo móvil a la etiqueta NFC	
	5		El sistema leerá la etiqueta y junto con el token de usuario los enviará al servidor donde una vez comprobados los datos, enviará una señal a la cerradura del objeto y permitirá acceso.
	6		El sistema mostrará un mensaje informando al usuario de que el control se ha llevado a cabo de forma correcta.
ESCENARIO ALTERNATIVO		ACTORES	SISTEMA
	1	El usuario seleccionará la opción “Control de acceso”	
	2	El usuario acercará el dispositivo móvil a la etiqueta NFC	
	3		El sistema comprobará si existe conexión con el servidor.
			El sistema mostrará un mensaje de error de conexión con el servidor.
POST-CONDICIÓN	La puerta o cerradura del objeto quedará liberado para acceso al usuario.		

Tabla 41. CU-005 - Control de acceso

4.4. Matriz de trazabilidad

A continuación se mostrará una matriz de trazabilidad para informar acerca de los casos de uso y los requisitos que afectan. Se dispondrá una matriz por cada tipo de requisito.

		Casos de uso				
		CU-001	CU-002	CU-003	CU-004	CU-005
Requisitos funcionales de software	RF-0001	X				
	RF-0002			X		
	RF-0003	X				
	RF-0004		X			
	RF-0005	X				
	RF-0006	X				
	RF-0007	X				
	RF-0008	X				
	RF-0009	X				
	RF-0010	X	X	X		
	RF-0011				X	
	RF-0012				X	
	RF-0013				X	
	RF-0014				X	
	RF-0015					
	RF-0016					X
	RF-0017					X
	RF-0018					X

Tabla 42. Matriz de trazabilidad requisitos func. Vs casos de uso.

	Casos de uso				
	CU-001	CU-002	CU-003	CU-004	CU-005

Requisitos no funcionales de software	RNF-0101					
	RNF-0102				X	X
	RNF-0103					
	RNF-0104					
	RNF-0105					
	RNF-0106	X	X	X	X	X
	RNF-0201	X	X	X	X	X
	RNF-0202	X	X	X	X	X
	RNF-0203					X
	RNF-0204					
	RNF-0301	X	X	X	X	X
	RNF-0302	X	X	X	X	X
	RNF-0303				X	X
	RNF-0304	X	X	X	X	X
	RNF-0305	X	X	X	X	X
	RNF-0306	X	X	X	X	X

Tabla 43. Matriz de trazabilidad requisitos no func. Vs casos de uso.

4.5. Conclusión

Vistos los casos de uso y los requisitos de software en esta sección, quedan claros los objetivos a alcanzar tras finalizar la fase de desarrollo.

Esta fase ha sido fruto de un análisis exhaustivo del sistema a desarrollar, y en ella se recogen aquellos aspectos que definen la funcionalidad y acabado del producto software objeto de este proyecto.

Sección 5

5. Diseño del sistema

5.1. Introducción

El diseño es una de las fases del ciclo de vida del desarrollo de un software. Dada la metodología de desarrollo usada en este proyecto, sólo se puede comenzar con la fase de diseño una vez finalizada la de análisis.

En esta fase se va a explicar la arquitectura lógica y física del sistema. De la arquitectura lógica se expondrá diseño estático y dinámico así como el diseño de la base de datos.

5.2. Arquitectura física

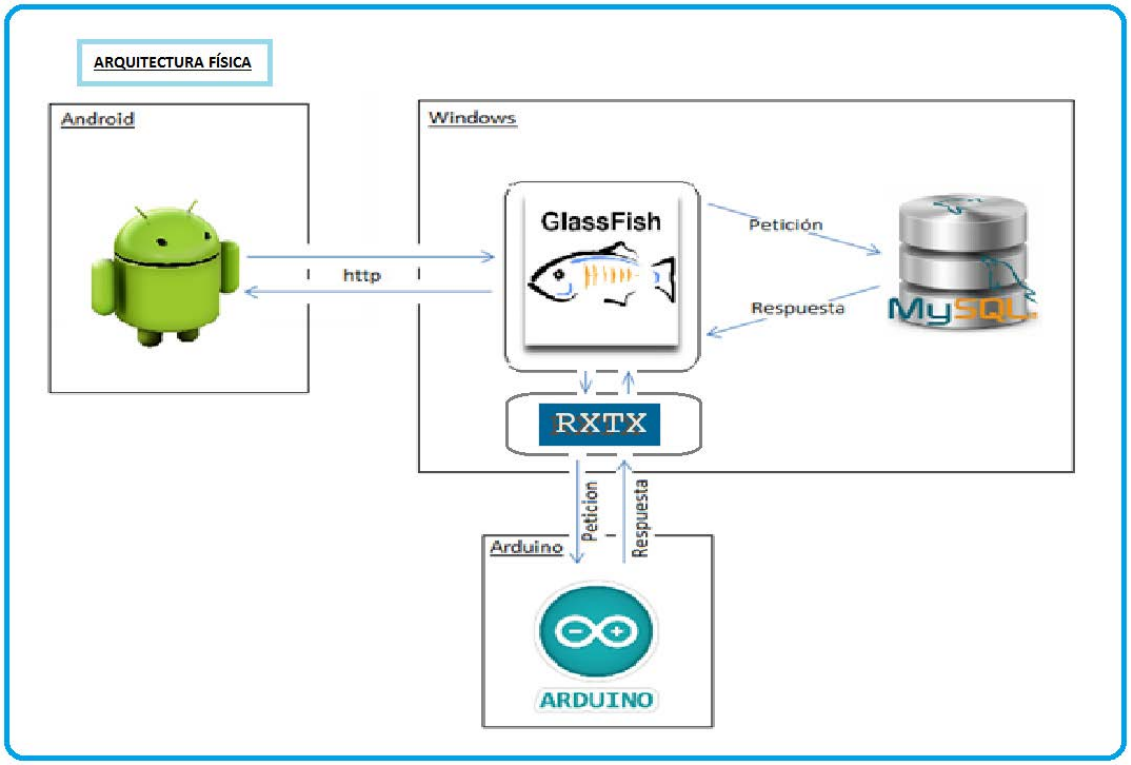


Figura 10: Arquitectura física

Para poder implantar el proyecto propuesto hacen falta los siguientes componentes:

- Teléfono móvil Android.
- Servidor de aplicaciones GlassFish.
- Sistema gestor de base de datos MySQL.
- Arduino Mega o similar.

La aplicación Android sólo se ejecutará sobre un teléfono Android. Los servicios alojados en el servidor de aplicaciones GlassFish se ejecutarán sobre el sistema operativo Windows. La base de datos se ejecutará sobre el mismo servidor que el servidor GlassFish aunque no tiene por qué ser así, de hecho, lo más habitual es que la base de datos se encuentre alojada en otro servidor distinto. Por último, tenemos la plataforma Arduino que es independiente ya que se precisa un hardware específico independiente e irá conectado al servidor Windows mediante la conexión USB.

La Aplicación central que se encargará de gestionar la información que llegue desde la aplicación móvil y realizar las distintas comprobaciones se ejecutará sobre el servidor de aplicaciones GlassFish. La aplicación almacenará y recuperará los datos por medio del SGBD MySQL.

Tal y como se puede ver en el gráfico de arriba, se observar que la aplicación alojada en el servidor GlassFish es el corazón del sistema. La aplicación Android es la que inicia cualquier proceso en el servidor por medio de conexión a través del protocolo http, éste hace la gestión pertinente y es el que se comunica con la base de datos y Arduino si procede, en caso de ser necesario devuelve el resultado de las operaciones a la aplicación móvil.

5.3. Arquitectura lógica

En este apartado se va a proceder a realizar una descripción general de la arquitectura lógica del sistema. Se dará a conocer el funcionamiento global del mismo, dejando para los siguientes apartados la descripción lógica de cada uno de los módulos que se muestran a continuación. Con esta descripción se pretende mostrar de una forma clara los distintos módulos que intervienen y las interacciones entre cada uno de ellos.

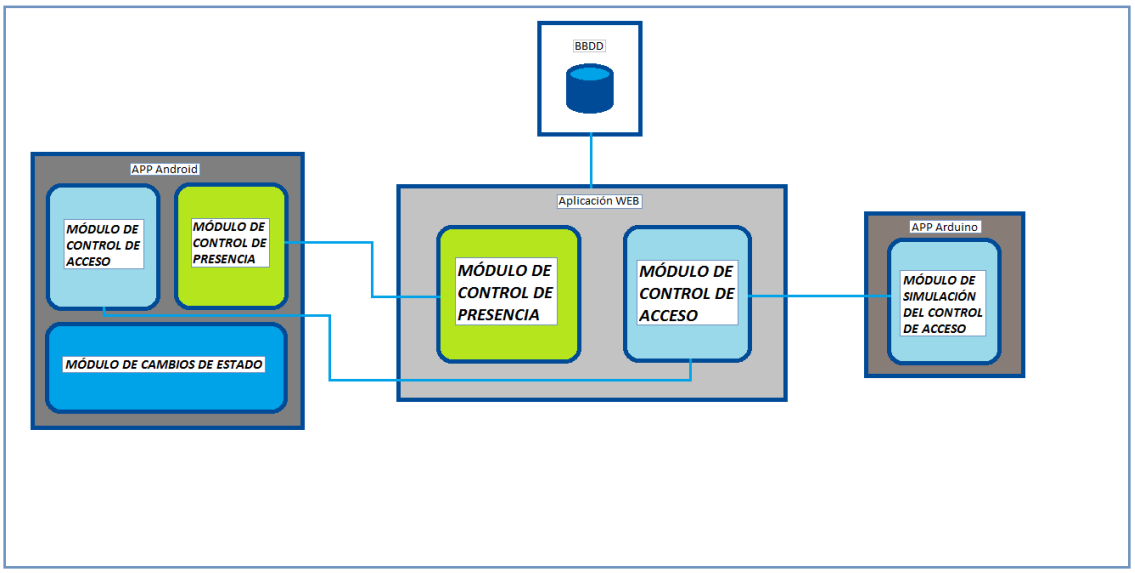


Figura 11: ARQUITECTURA LÓGICA

A continuación, se procederá a explicar brevemente cada uno de los módulos representados en el esquema anterior y la comunicación o interacción con otros.

De un vistazo rápido al esquema, se puede determinar que existen tres módulos en la parte de la aplicación móvil, algunos de ellos replicados en la aplicación alojada en el servidor y otros no. Esto se debe a que en el servidor deben existir las funcionalidades para procesar las operaciones que vienen desde la aplicación móvil.

Módulo de cambio de estados: se puede observar que el módulo de cambio de estados tiene un funcionamiento local, es decir, no necesita comunicación alguna con el servidor.

Como ya se ha comentado en apartados anteriores, para cambiar el estado del teléfono sólo hace falta una etiqueta NFC con el código bien definido correspondiente a un estado y la aplicación móvil aplica dicho estado. Este módulo consta de dos sub-módulos: el primero consiste en definir los estados y grabarlos en una etiqueta, y, el segundo es para aplicar los estados grabados en una etiqueta NFC leyendo el contenido de la misma.

Módulo de control de acceso: tal y como se ve en el esquema de arriba, este módulo se comunica con el correspondiente en el servidor para determinar si la persona que ha solicitado el acceso tiene o no permisos para llevar a cabo esta acción. La información correspondiente al control de acceso se aloja en la base de datos, por lo que el módulo de control de acceso en el servidor se comunica con la base de datos. Una vez se verifique que el usuario es válido (tiene permisos para acceder) el módulo de control de acceso del servidor envía una señal al simulador haciendo que se encienda un LED varias veces simulando la apertura de una puerta, caja fuerte, etc. Para determinar los permisos del usuario del teléfono móvil, se envía el código de enlace del móvil con la cuenta de usuario alojada en el servidor, los datos del usuario nunca se envían hacia la aplicación móvil.

Módulo de control de presencia: este módulo registra la fecha y la hora, y las coordenadas de GPS actuales en el sistema en base a un usuario determinado. Para determinar de qué usuario se trata, este módulo se comunica con el servidor enviándole un código único (código de enlace del móvil con cuenta de usuario). El módulo de control de presencia alojado en el servidor consulta la base de datos y obtiene el usuario si existe. En el caso de ser válido graba la información correspondiente en dicha base de datos, de lo contrario devuelve un error a la aplicación móvil.

5.4. Diseño de componentes

5.4.1. Módulo de cambios de estado

Este módulo se encarga de cambiar los estados del teléfono a partir de una etiqueta NFC. Tal y como se ha explicado en apartados anteriores, este módulo se encargará tanto de generar el código de modo como de aplicarlo. A continuación se muestran los diagramas de clases que intervienen en este proceso así como una breve explicación de cada una de ellas:

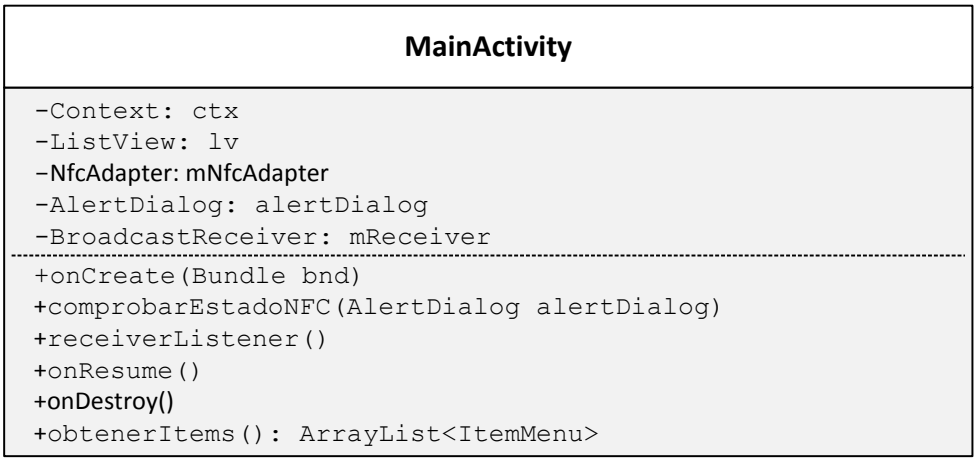


Figura 12. Clase MainActivity

Esta es la clase encargada de cargar la pantalla principal de la aplicación, a partir de ella se accederá a las distintas funcionalidades. Tiene como atributos el contexto (necesario para compartir contenido entre distintas clases tipo Activity en Android), una lista con los distintos menús de la aplicación, el dispositivo NFC (es necesario para inicializarlo y definir un escucha para la lectura y escritura) y una ventana de notificación para mostrar los distintos mensajes de aviso, información y errores, y que será compartida entre las distintas pantallas de la aplicación.

Algunos métodos son nativos, es decir, heredados de la clase Activity de Android y con el resto se precisan para obtener la lista de las opciones de menú y un método para comprobar el estado de NFC del dispositivo móvil.

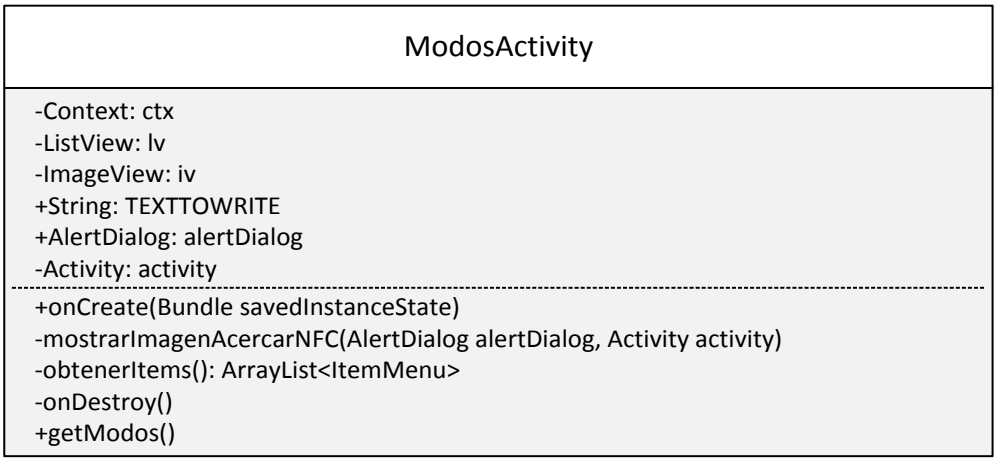


Figura 13. Clase ModosActivity

Una vez seleccionada la opción de “definir modos”, se presenta la siguiente pantalla en donde se muestran los distintos modos y la opción de crear un modo nuevo. La clase ModosActivity representa la interfaz gráfica de la pantalla que se ha mencionado, y como se puede observar tiene un atributo llamado TEXTTOWRITE que contendrá el código del modo elegido para grabar en una etiqueta. El método “mostrarImagenAcercarNFC” se invoca cuando se elige el modo y se selecciona la opción de guardar, la imagen que se muestra ayuda al usuario a continuar correctamente con el proceso para el guardado del dato en la etiqueta NFC. El atributo “iv” es utilizado para mostrar un icono que cuando se pulsa se cierra la pantalla actual y se muestra la anterior.

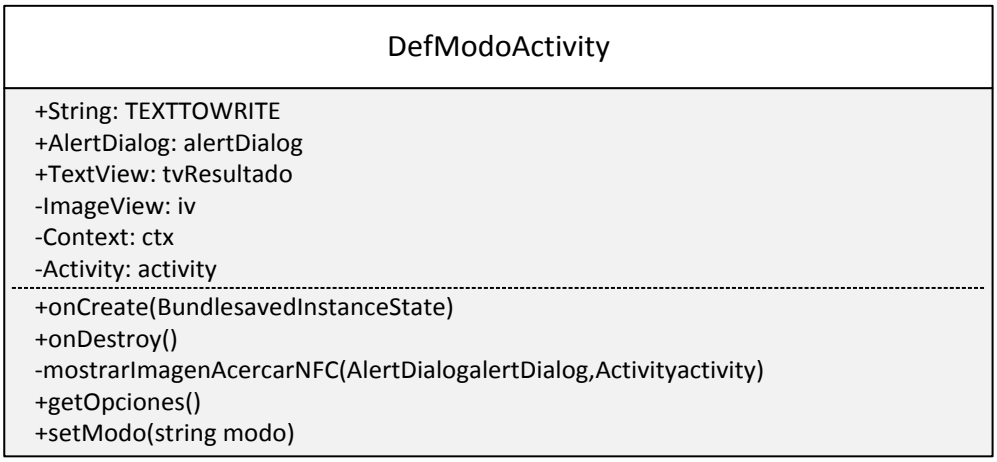


Figura 14. Clase DefModoActivity

Si se elige “definir nuevo modo” en vez de seleccionar uno predefinido, se llamará a esta clase que entre otras cosas que hace, cargará la interfaz para confeccionar un modo nuevo combinando distintos elementos como Wifi, Bluetooth, datos, sonido, etc. esta clase muestra las distintas opciones a seleccionar y posteriormente, una vez terminado el proceso de selección, permite guardar el modo confeccionado en una etiqueta NFC seleccionado la opción de guardar. El atributo “tvResultado” es usado para mostrar el resultado del guardado de datos que se ha seleccionado. “iv” es un atributo que se usa para mostrar el icono para volver atrás.

A continuación se mostrará y describirá cada uno de los diagramas de secuencia que intervienen en el funcionamiento de este módulo. El diagrama de secuencia tiene como

objetivo complementar al de clases para dar una idea de funcionamiento más completa del componente.

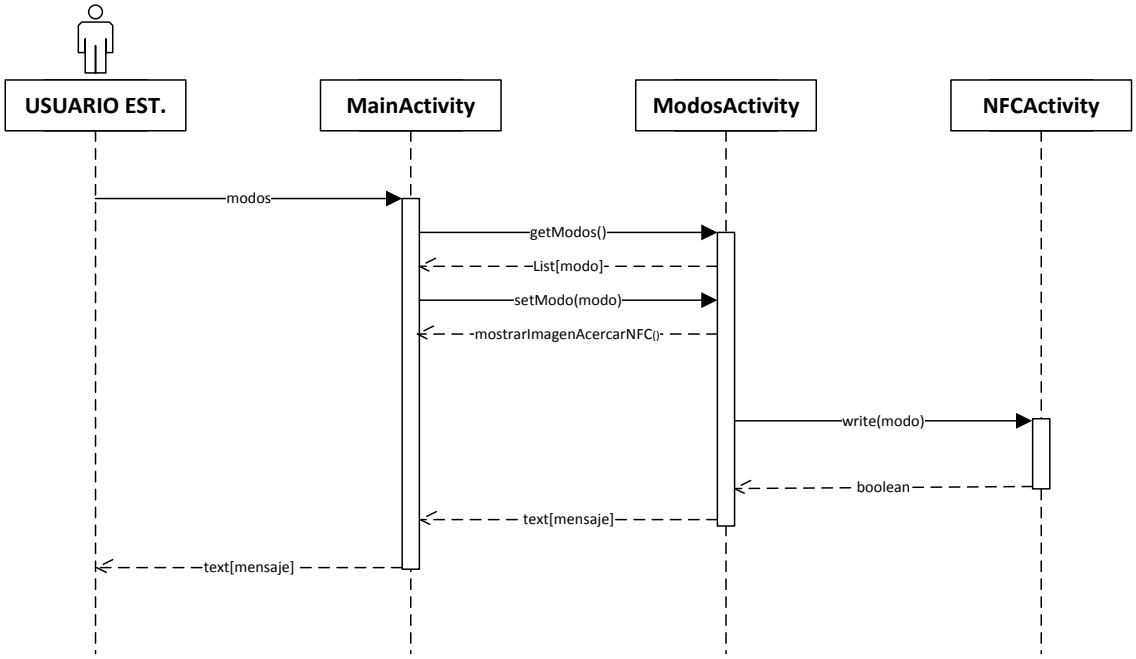


Figura 15. Diagrama de secuencia - Creación de un modo predefinido.

El actor representa a un usuario estándar (no registrado), este actor sólo tiene acceso a la parte de cambios de estado del teléfono. El actor, usuario estándar, puede definir y guardar un modo. También cabe señalar que el actor usuario registrado también puede realizar las mismas acciones.

Para crear un modo predefinido, el actor interacciona con la pantalla principal de la aplicación seleccionando la opción “definir modos”. Mediante la creación de la instancia de la clase ·ModosActivity” se tiene acceso a la pantalla donde se muestran los distintos modos y la opción de crear uno nuevo. Seleccionado el modo se muestra una imagen que indica al usuario el modo de proceder para guardar el contenido y se invoca el método write(string) para guardar el código correspondiente al modo elegido. Una vez se haya guardado el contenido o terminado de ejecutar el método write(string) éste devolverá el resultado de esta ejecución a la clase que lo ha invocado y ésta preparará un mensaje para mostrar finalmente en la ventana de notificación en la pantalla principal.

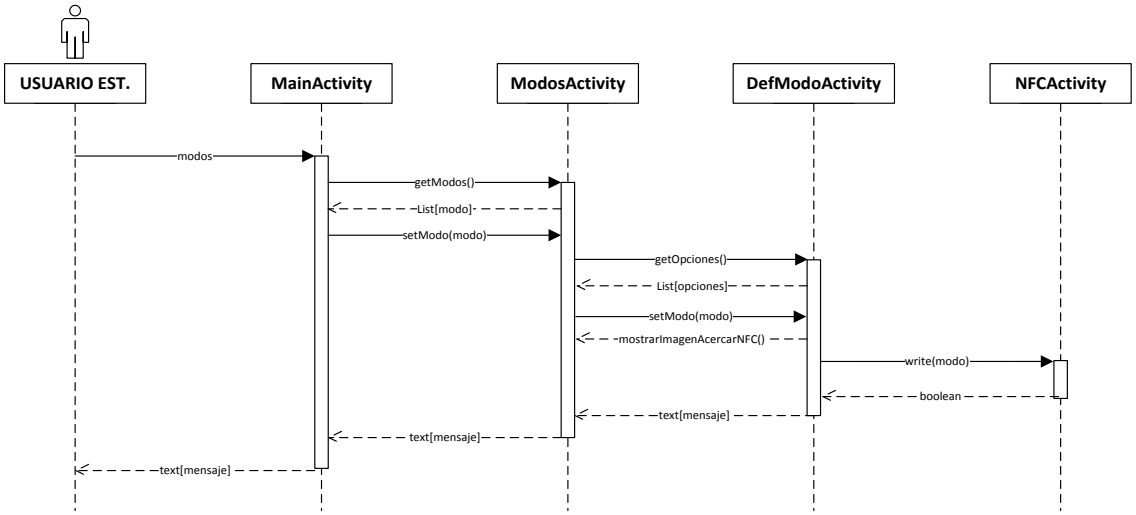


Figura 16. Diagrama de secuencia - Creación de un modo nuevo.

Al igual que en el caso anterior, el actor puede querer confeccionar un modo nuevo combinando distintos elementos como ya se ha comentado anteriormente. Para confeccionar un nuevo modo, el actor interacciona con la clase MainActivity, esta crea una instancia de la clase ModosActivity. Una vez el actor seleccione la opción de nuevo modo, la instancia u objeto de la clase ModosActivity creará un objeto de la clase DefModoActivity, mostrará las distintas opciones y finalmente cuando se seleccionen las opciones correspondientes invocará el método write de la clase NFCActivity para guardar el resultado en una etiqueta NFC. La clase NFCActivity encaminará un mensaje hasta la clase principal para mostrar el estado del resultado.

5.4.2. Módulo de control de presencia

Control de presencia es un módulo que requiere (ver apartado 5.3 arquitectura lógica) conectar el componente albergado en el dispositivo móvil con el del servidor. Para satisfacer esta necesidad se expondrán a continuación las clases que intervienen en este proceso y comentando aquellas que no se hayan comentado previamente.

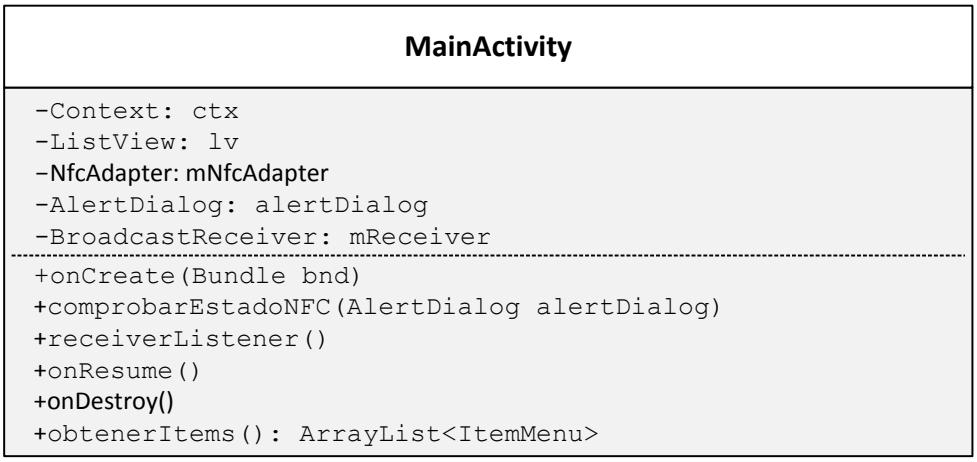


Figura 17. Clase MainActivity

Se recuerda que es la clase que invocará el módulo de control de acceso dentro de la parte móvil.

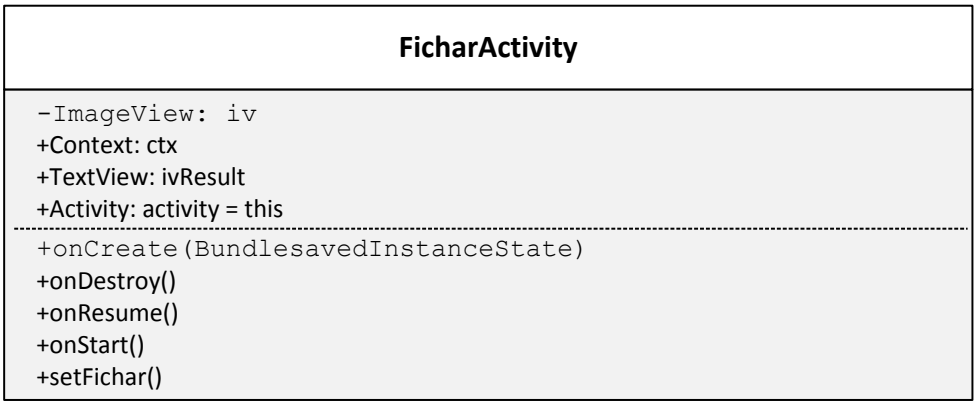


Figura 18. Clase FicharActivity

Una vez se seleccione la opción correspondiente a control de presencia en la clase MainActivity, se invocará esta clase y es ésta la encargada de conectar con el servicio web para llevar a cabo dicho control de presencia.

Los atributos “iv” y “ivResult” son a destacar, ya que los otros son similares a los que ya se han comentado en otras clases y cumplen la misma funcionalidad. “iv” es el atributo que almacenará el icono para volver atrás cerrando la pantalla actual. “ivResult” se usará para mostrar el resultado de la operación.

El método a destacar de esta clase es “setFichar()” es invocado una vez se comprueba en el constructor de la clase “onCreate(…)” que se puede establecer una conexión con el servicio web. “setFichar()” llama al servicio web pasando por parámetro el token, el código leído por NFC y las posiciones GPS como cadena de texto separando las coordenadas geográficas con “:”.

Figura 19. Clase ConexionEstado

Se puede observar que esta clase es muy sencilla ya que sólo dispone de un método y no tiene atributos. “getEstado()” llama al servicio web y comprueba si existe respuesta por parte de éste, en caso afirmativo devuelve estado true y en el otro caso devuelve false. Esta comprobación se hace siempre que se precise conexión con el servicio web y sirve para informar al usuario en caso de que no haya conexión disponible para comprobar el estado de ésta.

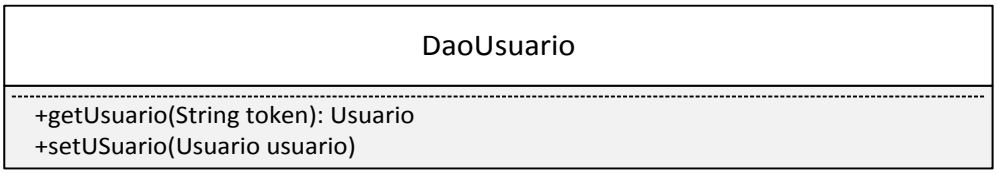


Figura 20. Clase DaoUsuario

La clase “DaoUsuario” es la encargada de acceder a la base de datos y obtener los datos de los usuarios del sistema. Se puede ver que la clase es muy sencilla ya que dispone de dos métodos “getUsuario(…)” y “setUsuario(…)”. Con esta clase se pretende separar la capa de datos del modelo.

“getUsuario(String token) devuelve un usuario en base a un token (código único para cada usuario) y null en caso de que resulte que el usuario no exista en la base de datos. Este método crea un objeto de tipo “Usuario” cuando obtiene los datos desde la base de datos.

"setUsuario(Usuario usuario)" se encarga de guardar en la base de datos el usuario que se le pasa por parámetro. Este método es necesario para la parte de administración.

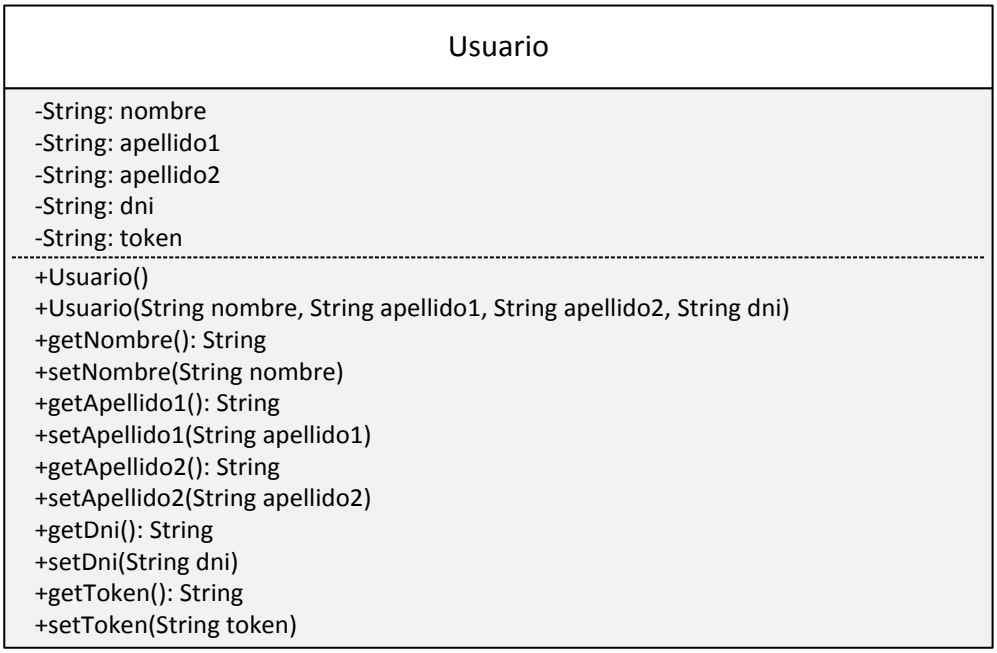


Figura 21. Clase Usuario

Usuario es la clase que alberga los datos del usuario registrado en el sistema y como se puede ver, tiene los atributos necesarios para identificar a un usuario y los métodos para obtener y sobrescribir el valor de cada uno de los atributos ya que éstos no son accesibles desde fuera de la clase. Esta clase pertenece a la capa del modelo de la aplicación.

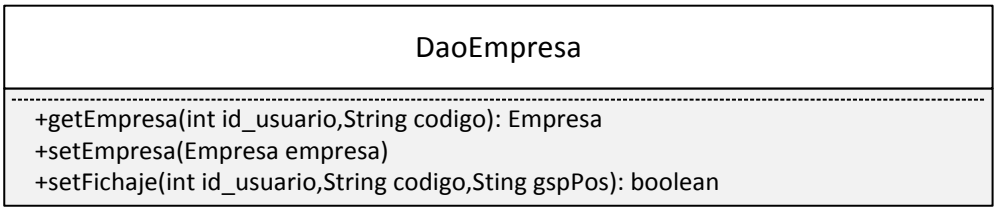


Figura 22. Clase DaoEmpresa

Al igual que la clase "DaoUsuario", la clase "DaoEmpresa" tiene el objetivo de acceder a los datos de las empresas almacenado en la base de datos. Se observa que no tiene atributos y sus métodos son públicos.

“getEmpresa(…,…)” es un método que recupera los datos de una empresa en función de los parámetros “id_usuario” y “código”. Si el usuario no pertecene a la empresa cuyo código es el parámetro que recibe el método o no existe alguno de los dos parámetros el resultado será null y si existe devolverá un objeto de tipo “Empresa”

“setEmpresa(…)” se encarga de guardar los datos de una empresa en la base de datos.

“setFichaje(…,…,…)” dados el “id_usuario”, el “código” y las coordenadas gps “gpsPos” graba en base de datos con fecha y hora actuales el fichaje del usuario. Antes de acceder a la base de datos y realizar el proceso, comprueba que el usuario y el código de empresa existen y se relacionan entre ellos y para ello invoca el método “getEmpresa(...,…)”.

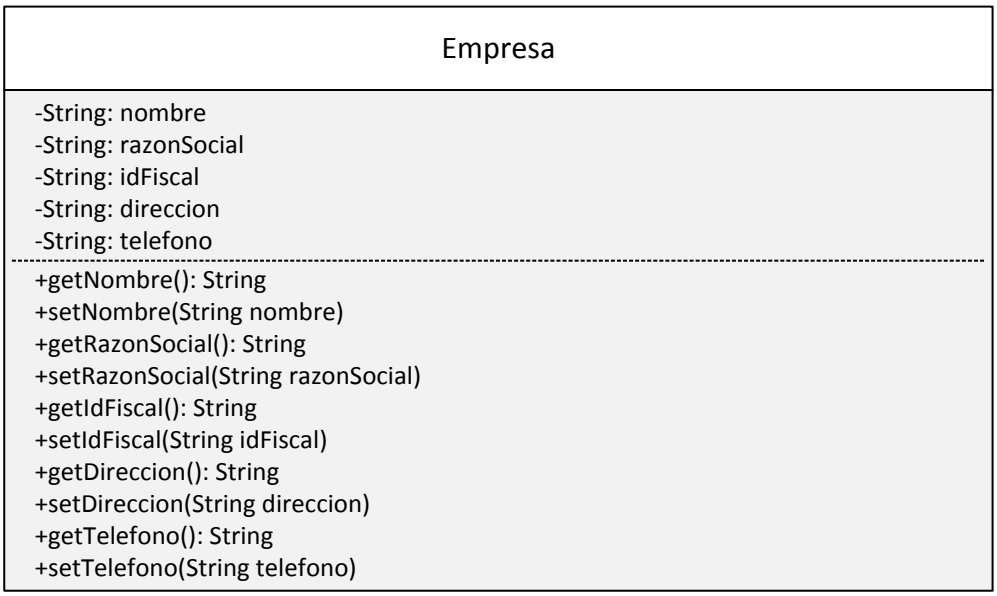


Figura 23. Clase Empresa

Esta clase representa conceptualmente una empresa, los atributos recogen los datos identificativos de cada empresa y como se puede observar son privados (ámbito de clase) y los métodos son para acceder y sobrescribir cada uno de los atributos. Es una clase que pertenece a la capa modelo de la aplicación.

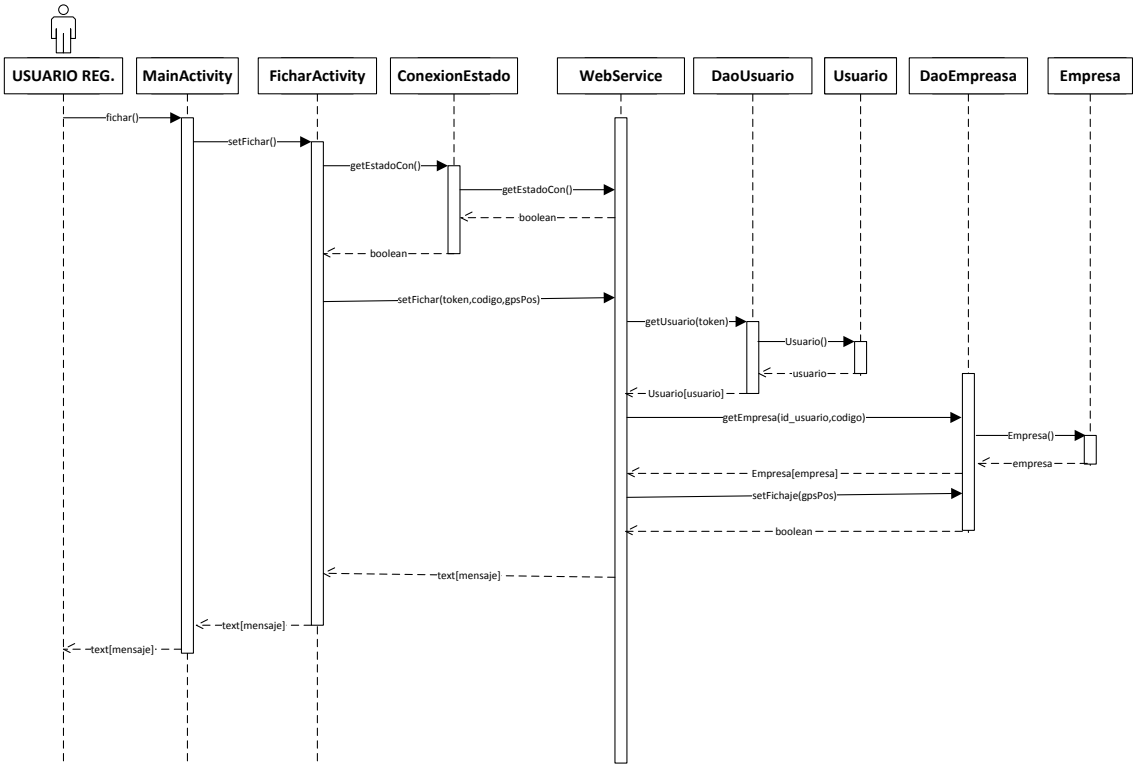


Figura 24. Diagrama de secuencia - Control de presencia.

Este diagrama de secuencia representa todas las interacciones entre las distintas clases para llevar a cabo control de presencia. El actor solicita realizar control de presencia desde la clase principal "MainActivity" y ésta crea una instancia de la clase "FicharActivity" que es la que realmente se encargará del proceso. La clase "FicharActivity" se encargará de comprobar la conexión con el servicio web por medio de la clase "EstadoConexion", si la conexión es correcta se proseguirá con el proceso comprobando el usuario y la empresa. Si los datos son correctos, se solicitará al método "setFichaje(…,…,…)" de la clase "DaoEmpresa" y este método devolverá el estado de ejecución de la tarea para informar al usuario.

Cabe destacar que en este proceso interviene únicamente el actor "USUARIO REGISTRADO".

5.4.3. Módulo de control de acceso

Al igual que en el caso del Módulo de control de presencia (ver apartado 5.4.2) el Módulo de control de acceso consta de un componente local y otro en el servidor (ver

apartado 5.3 arquitectura lógica). Para satisfacer esta necesidad se van a exponer a continuación las distintas clases que intervienen en este proceso comentando aquellas que son propias del proceso y que no se hayan comentado en apartados anteriores en esta sección.

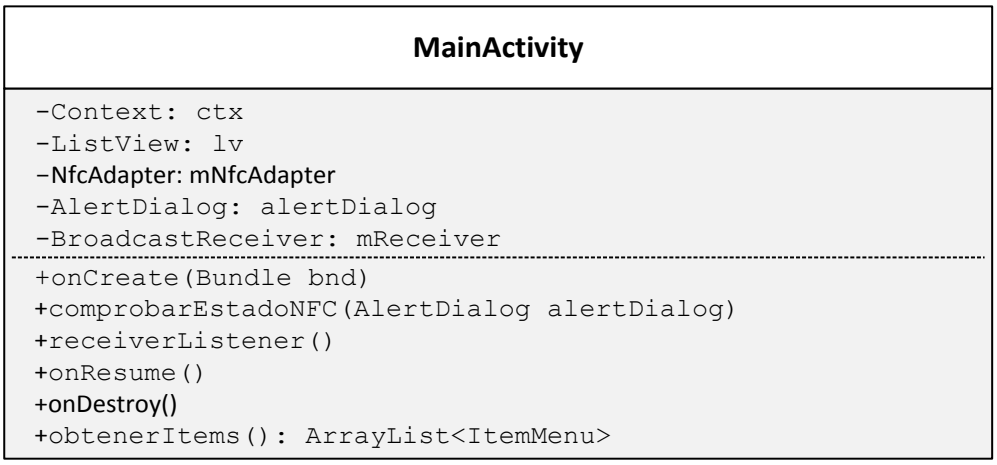


Figura 25. Clase MainActivity

Esta clase ya se ha comentado anteriormente, cumple las mismas funciones que las descritas en apartados anteriores.

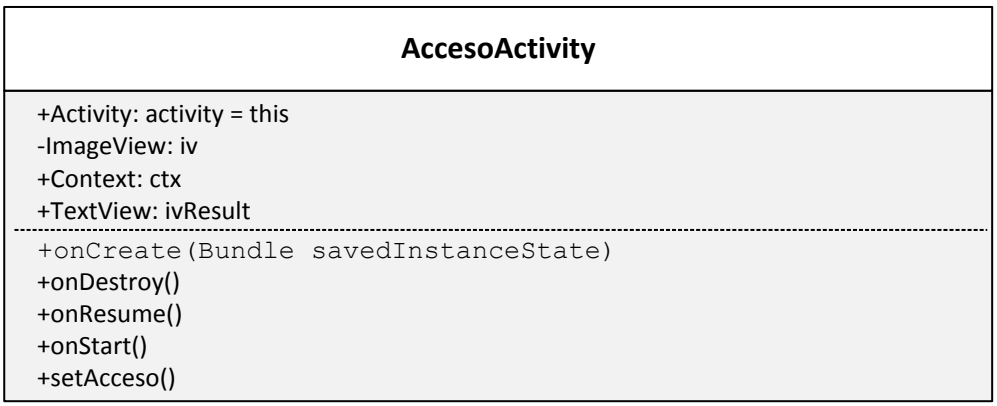


Figura 26. Clase AccesoActivity

Por medio de esta clase se canaliza la información hacia el servidor para realizar un control de acceso. Se observa la existencia de un método público llamado “setAcceso()” encargado de invocar al servicio web y solicitar acceso dados un “token “y un “código”. Antes de proceder a invocar la solicitud de acceso, la clase “AccesoActivity” comprobará la existencia de conexión con el servicio web.

Figura 27. Clase ConexionEstado

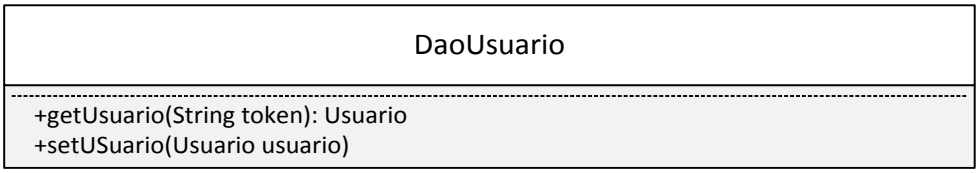


Figura 28. Clase DaoUsuario

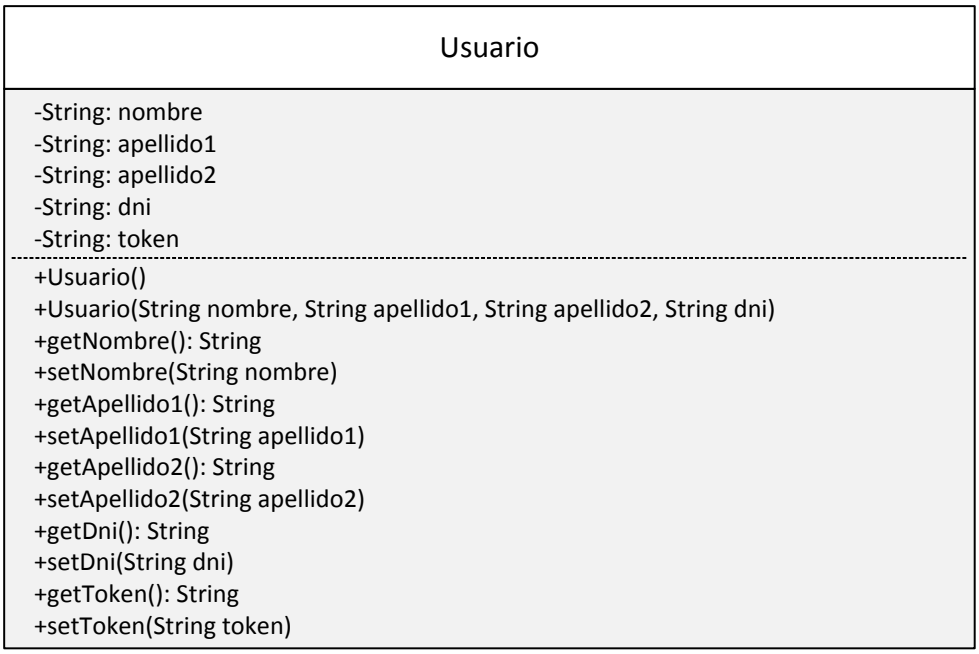


Figura 29. Clase Usuario

Las clases “ConexionEstado”, “DaoUsuario” y “Usuario” se mencionan por formar parte del proceso, estas clases se han comentado en apartados anteriores.

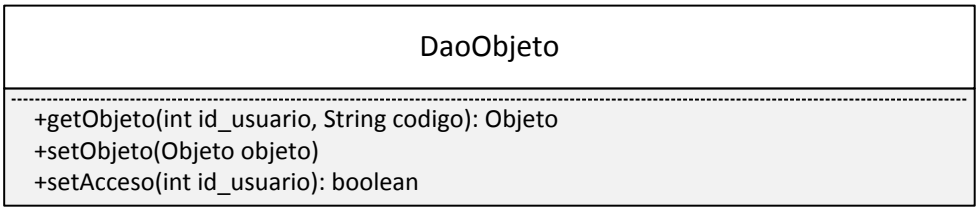


Figura 30. Clase DaoObjeto

“DaoObjeto” tiene como finalidad acceder a la base de datos (capa de acceso a los datos) y extraer la información de los objetos solicitados comprobando que el usuario que los

solicita tiene acceso a ellos. “setAcceso(…)” devuelve true en caso de cumplirse las condiciones y false en el caso contrario.

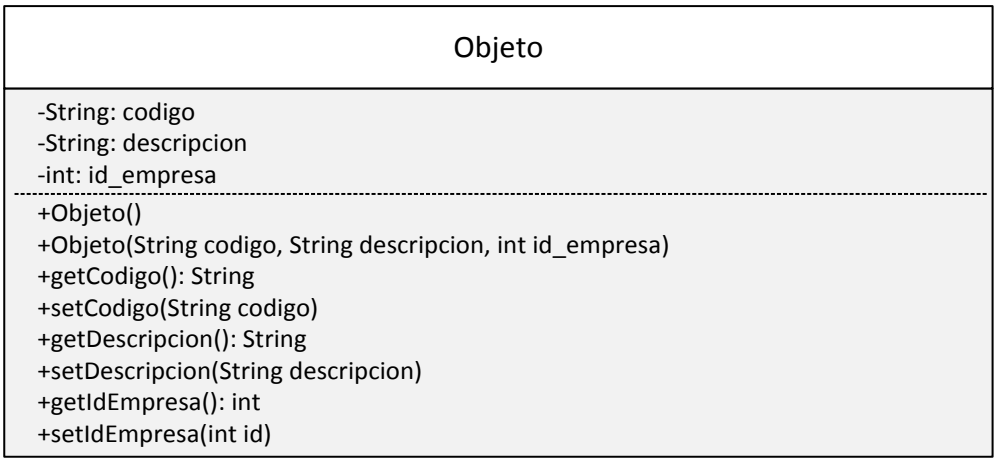


Figura 31. Clase Objeto

Esta clase representa aquellos objetos a los que se pueden acceder tales como puertas, cajas fuertes, candados, etc. el código –atributo de la clase “Objeto”- es un código único para cada objeto incluido en el sistema. Los métodos de esta clase únicamente se usan para obtener o sobrescribir cada uno de los atributos ya que éstos no son accesibles desde fuera de la clase.

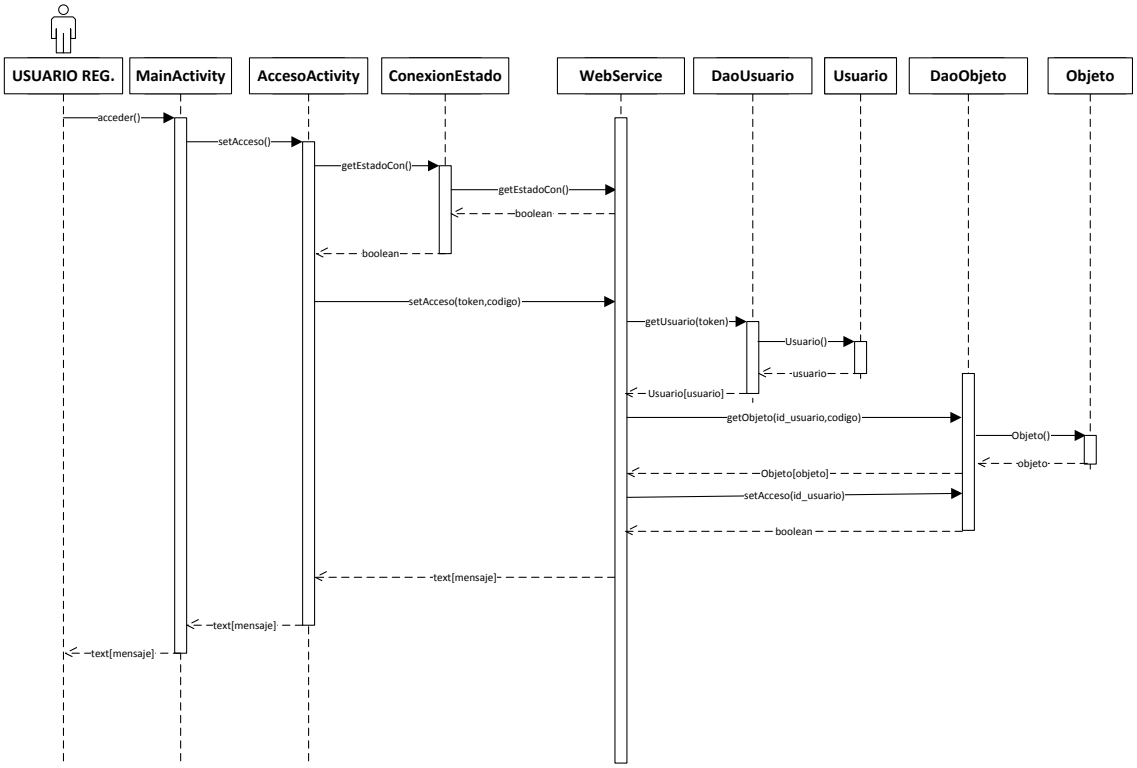


Figura 32. Diagrama de secuencia - Control de acceso.

Este diagrama de secuencia representa todo el proceso que existe desde que el usuario solicita realizar un control de acceso hasta completar el proceso. Se puede observar que el proceso es similar al de control de presencia, la diferencia fundamentalmente existe como se puede apreciar, en el servidor a la hora de acceder a realizar dicho control ya que sólo son necesarios dos parámetros: el token y el código del objeto. El usuario recibirá un mensaje con el resultado de esta operación.

5.5. Diseño de la base de datos

A continuación se va a proceder a describir la base de datos que usará la aplicación. Tal y como ya se ha comentado anteriormente, la aplicación registrará la presencia de los empleados al llegar o abandonar su puesto de trabajo, para ello se necesitará que la aplicación móvil se conecte al servidor central y verifique la identidad del usuario, así como posteriormente registrar dicha llegada o salida y hacer que estos datos sean persistentes.

Para el control de acceso, la aplicación debe verificar si el usuario tiene permisos suficientes para acceder al sitio solicitado ya que de lo contario no deberá dejar que acceda. Si el usuario tiene permisos, el servidor debe enviar una señal al mecanismo de la cerradura de la puerta para abrirla y permitir así el acceso.

Para cubrir las necesidades de la aplicación, se va a usar una base de datos y se diseñará para ello un conjunto de tablas relacionadas que contendrán toda la información necesaria para el correcto funcionamiento de la aplicación.

A continuación se muestra un diagrama Entidad Interrelación de la base de datos que representa una descripción conceptual de las relaciones entre las distintas entidades involucradas.

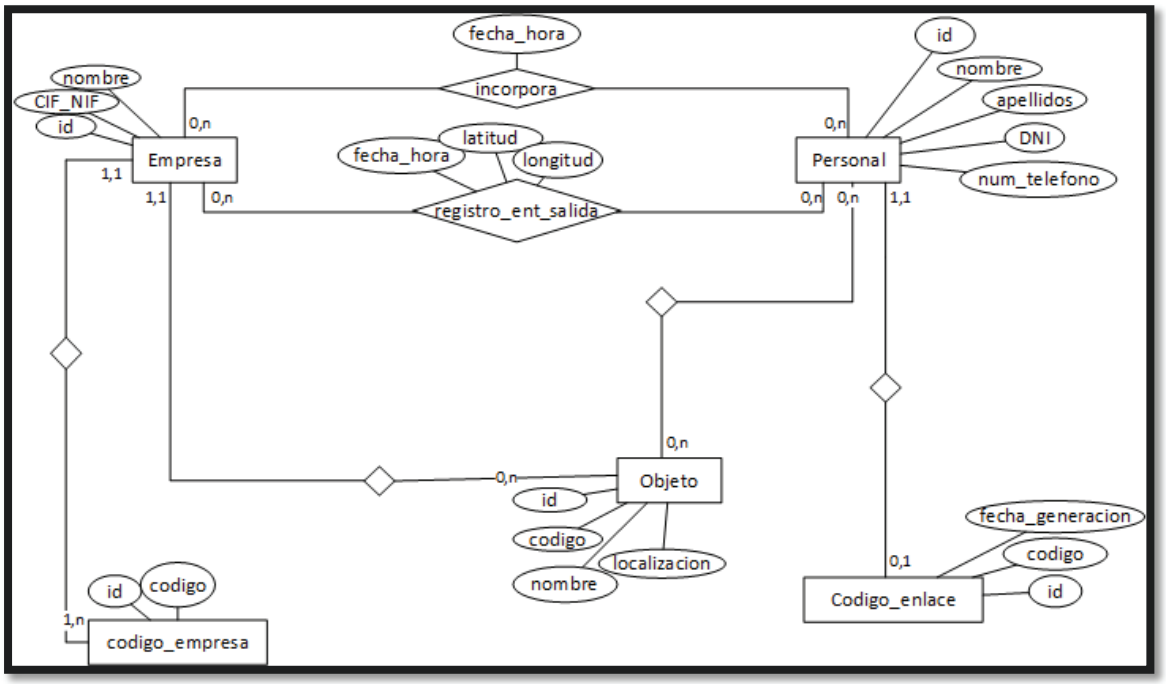


Figura 33: Diagrama Entidad Interrelación

El diagrama ER anterior se transforma en el siguiente esquema físico de base de datos:

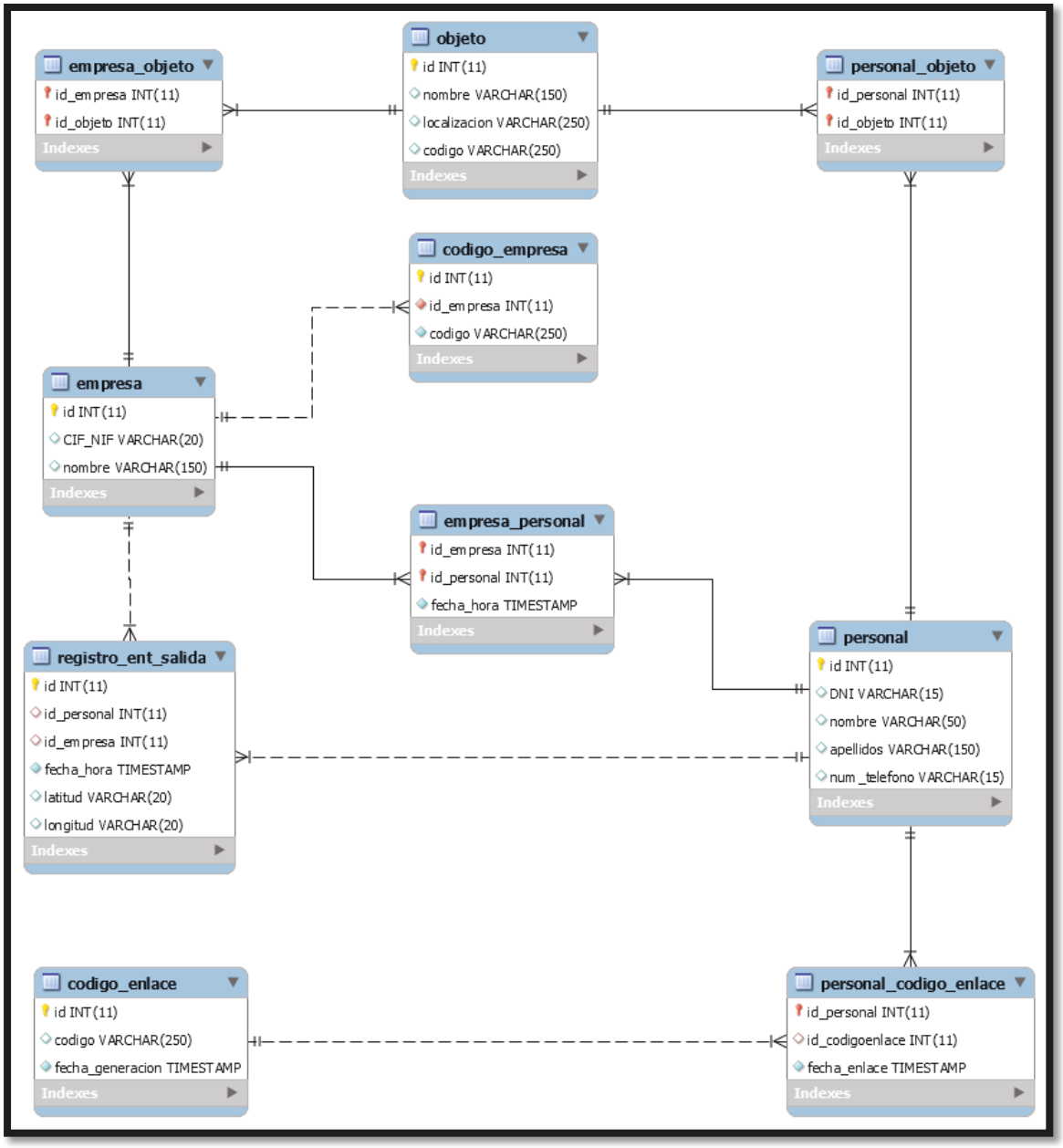


Figura 34: *Esquema ER de la base de datos*

Finalmente se obtiene el esquema físico para crear las tablas de la base de datos.

```
CREATE TABLE empresa
(
       id int primary key
      ,CIF_NIF varchar(20)
      ,nombre varchar(150)
);

CREATE TABLE codigo_empresa
(
       id int primary key
      ,id_empresa int not null
      ,codigo varchar(250) not null
      ,foreign key (id_empresa) references empresa (id)
);

CREATE TABLE objeto
(
       id int primary key
      ,nombre varchar(150)
      ,localizacion varchar(250)
      ,codigo varchar(250) unique
);

CREATE TABLE empresa_objeto
(
       id_empresa int
      ,id_objeto int unique
      ,primary key (id_empresa, id_objeto)
      ,foreign key (id_empresa) references empresa(id)
      ,foreign key (id_objeto) references objeto(id)
);

CREATE TABLE personal
(
      id int primary key
      ,DNI varchar(15) unique
      ,nombre varchar(50)
      ,apellidos varchar(150)
      ,num_telefono varchar(15)
);

CREATE TABLE codigo_enlace
(
      id INT primary key
      ,codigo varchar(250) unique
      ,fecha_generacion timestamp
);

CREATE TABLE personal_codigo_enlace
(
       id_personal int primary key
      ,id_codigoenlace int unique
      ,fecha_enlace timestamp
      ,foreign key (id_personal) references personal (id)
      ,foreign key (id_codigoenlace) references codigo_enlace (id)
);
```

```
CREATE TABLE empresa_personal
(
        id_empresa int
       ,id_personal int
       ,fecha_hora timestamp
       ,primary key (id_empresa, id_personal)
       ,foreign key (id_empresa) references empresa (id)
       ,foreign key (id_personal) references personal (id)
);

CREATE TABLE personal_objeto
(
        id_personal int
       ,id_objeto int
       ,primary key (id_personal, id_objeto)
       ,foreign key (id_personal) references personal (id)
       ,foreign key (id_objeto) references objeto(id)
);

CREATE TABLE registro_ent_salida
(
        id int auto_increment primary key
       ,id_personal int
       ,id_empresa int
       ,fecha_hora timestamp
       ,latitud varchar(20)
       ,longitud varchar(20)
       ,unique (id_personal, fecha_hora)
       ,foreign key (id_personal) references personal (id)
       ,foreign key (id_empresa) references empresa (id)
);
```

Figura 35: Esquema físico de base de datos

5.6. Conclusión

En esta sección se han tomado las decisiones que se han visto oportunas para confeccionar el sistema que se requiere tras conocerse la fase de análisis. Esta fase representa el punto clave para decidir qué aspecto va a tener la aplicación y sobre todo cómo van a interactuar sus componentes y éstos con los externos. En definitiva se puede decir que lo habitual en esta fase es que el analista tome las riendas y decida el funcionamiento de la aplicación teniendo en cuenta los requerimientos del cliente o usuario que ha solicitado esta aplicación.

Para realizar un buen diseño de software el analista debe cumplir con algunos requisitos fundamentales; en primer lugar debe conocer bien lo que el cliente quiere y en segundo lugar debe conocer bien la tecnología que se va a usar para confeccionar dicha aplicación.

Dada la importancia que tienen estos dos aspectos mencionados, lo habitual cuando se ha alcanzado esta fase es tener bien claros los requisitos del usuario y el cómo plantear la aplicación, ya que de lo contrario sería muy costoso en tiempo y dinero rectificar.

Tras concluir esta fase del proyecto, se procederá con el desarrollo. En esta memoria no se va a exponer dicha fase. La siguiente fase a tratar en esta memoria es la de pruebas, en ella se tratarán las pruebas realizadas para validar el correcto funcionamiento del sistema y evitar así entregar un producto con posibles errores.

Sección 6

6. Pruebas

6.1. Introducción

Para asegurar el correcto funcionamiento del sistema se ha elaborado un plan de pruebas que servirá, además de para detectar y corregir errores, para verificar que el software hace lo que se ha descrito en la fase de análisis. La correcta ejecución de estas pruebas significa que el software cumple correctamente todos los requisitos que se han definido y que han dado lugar a su desarrollo.

Las pruebas que se describen en esta sección son pruebas funcionales, es decir, se prueban funcionalidad por funcionalidad hasta completar el proceso. La ejecución de estas pruebas se lleva a cabo de forma manual, así como, el resultado también se verifica de forma manual siguiendo el plan de pruebas que se describe a continuación en el siguiente apartado.

Estas pruebas se han diseñado en base a los requisitos funcionales definidos en la fase de análisis. La ejecución de las mismas se ha llevado a cabo tras finalizar cada una de las funcionalidades. En el caso de que se detectara algún error durante la ejecución de la prueba, se procede a repasar dicha funcionalidad y corregir dicho error, se considera que una prueba ha sido ejecutada con éxito cuando se obtiene por salida el resultado esperado.

6.2. Plan de pruebas

Tal y como se ha comentado en la introducción de esta sección, para comprobar y verificar que todos los requisitos de software son satisfechos se ha definido un conjunto de pruebas. Cada prueba representa la comprobación y verificación de cada requisito.

Cada prueba es identificada por un código único, descripción breve, un conjunto de datos de entrada y otro de salida o resultado esperado. A continuación se representa en una tabla a modo de ejemplo una prueba funcional.

Identificador prueba	**Descripción**	
Requisito referencia		
Entrada		***Salida***
Paso 1	Datos de entrada 1	Datos de salida 1
Paso 2	Datos de entrada 2	Datos de salida 2
…	…	…

Tabla 44. Descripción de una prueba

Como los requisitos no funcionales no se recogen en las pruebas, se repasará cada uno de ellos y se validará su cumplimiento de forma manual.

6.3. Pruebas

PF-0001	**Definir un modo predeterminado y guardarlo en una etiqueta NFC**	
RF-0001		
Entrada		***Salida***
Paso 1	Abrir la aplicación	Se abre la aplicación y aparece el menú principal.
Paso 2	Seleccionar la opción de "Definir modos"	Se abre una nueva ventana con varias opciones para seleccionar.
Paso 3	Seleccionar uno de los modos	Aparece un mensaje para proceder a guardar el modo en una etiqueta NFC.
Paso 4	Acercar el Smartphone a una etiqueta NFC.	Aparece un mensaje indicando si se ha guardado o no el mensaje.

Tabla 45. PF-0001 - Definir un modo predeterminado y guardarlo en una etiqueta NFC

PF-0002	**Aplicar un modo a partir de una etiqueta NFC**	
RF-0002		
Entrada		***Salida***
Paso 1	Desbloquear el Smartphone si estuviera con la pantalla bloqueada.	Pantalla desbloqueada.

Paso 2	Acercar el Smartphone a una etiqueta NFC.	Aparece una notificación en la parte superior de la pantalla indicando que "la aplicación NFCInteractive ha cambiado el estado del teléfono" y si el modo fuera distinto al que ya estaba aplicado se vería este cambio aplicado.

Tabla 46. PF-002 - Aplicar un modo a partir de una etiqueta NFC

PF-0003	**Definir modos predeterminados**	
RF-0003 , RF-0005 , RF-0006 , RF-0007 , RF-0008 y RF-0009		
Entrada		***Salida***
Paso 1	Abrir la aplicación "NFCInteractive"	La aplicación se abre y aparece el menú principal.
Paso 2	Seleccionar la opción "Definir modo".	Aparece una ventana con un listado con los siguientes modos predeterminados: • Normal • Avión • Reunión • Coche • Casa
Paso 3	Seleccionar "Normal"	Aparecen indicaciones para guardar el estado en un etiqueta NFC
Paso 4	Acercar el Smartphone a una etiqueta NFC.	Desaparecen las indicaciones para guardar el modo y aparece el resultado de la operación indicando si se ha guardado o no dicha información.

Tabla 47. PF-0003 - Definir modos predeterminados

PF-0004 **RF-0004**	**Definir un modo nuevo**	
Entrada		***Salida***
Paso 1	Abrir la APP	Aparece el menú principal de la app
Paso 2	Seleccionamos la opción "Definir modos"	Aparece un listado de los modos predeterminados y el último elemento de la lista es "Definir nuevo"
Paso 3	Seleccionamos la opción "Definir nuevo"	Aparece un listado de los elementos a configurar para definir un modo. • Modo audio • Wifi • Bluetooth • Datos • GPS
Paso 4	Seleccionamos el botón "Guardar"	Aparecen indicaciones a seguir para guardar el modo definido en una etiqueta NFC.
Paso 5	Acercar el Smartphone a una etiqueta NFC	Desaparecen las indicaciones y aparece el resultado de esta operación.

Tabla 48. PF-0004 - Definir un modo a medida

PF-0005 **RF-0010**	**Registrar la presencia de una persona a través del dispositivo móvil.**	
Entrada		***Salida***
Paso 1	Abrir la APP "NFCInteractive"	Aparece el menú principal de la APP "NFCInteractive".
Paso 2	Seleccionamos la opción "Control de presencia"	Se abre una nueva ventana y seguidamente aparecen las indicaciones para llevar a cabo el registro de presencia.

Paso 3	Acercamos el Smartphone a la etiqueta NFC (etiqueta NFC con información de la empresa o entidad que se quiere registrar).	Aparece un mensaje con el resultado del registro.

Tabla 49. PF-0005 - Registrar la presencia de una persona a través del dispositivo móvil.

PF-0006 RF-0012	**Acceso a zonas o elementos restringidos mediante el dispositivo móvil**	
Entrada		***Salida***
Paso 1	Abrir la APP "NFCInteractive"	Aparece el menú principal de la APP.
Paso 2	Seleccionamos la opción "Control de acceso"	Se abre una nueva ventana y seguidamente aparecen las indicaciones para llevar a cabo el control de acceso.
Paso 3	Acercamos el Smartphone a la etiqueta NFC (etiqueta NFC con información del lugar o elemento al que se desea acceder).	Aparece un mensaje con el resultado de la operación.

Tabla 50. PF-0006 - Acceso a zonas o elementos restringidos mediante el dispositivo móvil

Con las pruebas que se han mencionado quedan probadas todas las funcionalidades del sistema dando por válidas todas ellas. Además se validan todos los requisitos de usuario "funcionales" contra el sistema concluyendo que el sistema se adapta al completo a los requisitos descritos en la fase de análisis.

En cuanto a los requisitos no funcionales, se han validado y probado durante la fase de desarrollo ya que suponen la base para comenzar a desarrollar el sistema, por ello se ha decidido no realizar la batería de pruebas al final de la fase de desarrollo dando por hecho que para superar esta fase se han tenido que cumplir tal y como se describieron en su momento.

6.4. Conclusión

Concluida la fase de pruebas y dado el visto bueno a las mismas supone que el sistema se puede poner a disposición del cliente con total fiabilidad, es decir, la última fase: Implantación.

Como ya se ha visto, se puede decir que esta fase es una de las más importantes en el desarrollo del software ya que el hecho de no realizar pruebas íntegras y exhaustivas del sistema conlleva a que éste no sea fiable. Durante la fase de pruebas se han detectado y corregido muchos errores que de no ser por estas pruebas se habrían arrastrado a la fase de implantación, sin embargo, se ha podido actuar a tiempo solucionándolos y se ha podido depurar el sistema a tiempo dando lugar a una versión mucho más estable.

La fase de pruebas es una manera que se tiene para demostrar tanto al equipo de desarrollo como al cliente que todos aquellos requisitos firmados con el cliente se han cumplido al 100%, si el cliente echa en falta alguna funcionalidad o simplemente discrepa con el equipo, sólo es cuestión de verificar si existe requisito alguno que contempla dicha funcionalidad y en ese caso recurrir al plan de pruebas para ver dónde se recoge y cómo funciona. Si se han hecho pruebas exhaustivas, se evitarán situaciones incómodas con el cliente.

Sección 7

7. Presupuesto

7.1. Introducción

En esta sección se va a proceder a realizar un presupuesto con los costes que ha supuesto el proyecto desde su comienzo hasta su finalización.

Se va a exponer en primer lugar el calendario laboral que se ha tomado como referencia para la realización del proyecto, tras esto se va a exponer el resumen de la planificación del proyecto comentando su duración dividida en fases. Al igual que esto, se incluirá el diagrama de Gantt donde se va a exponer de una forma gráfica el desarrollo del proyecto en sus distintas fases y cada una de las fases sus tareas más específicas. En los apartados posteriores se va a dar a conocer los costes de la realización del proyecto desglosado en diferentes tipos de coste. Y finalmente se va a proceder a resumir todos los costes comentados.

Para realizar este presupuesto se ha basado en la plantilla que ofrece la universidad como guía.

7.2. Calendario laboral

Para la realización de este proyecto se ha tomado como referencia un calendario laboral estándar, éste consta de cinco días laborables a la semana y de cinco horas cada día, lo que conforma un total de veinticinco horas semanales.

7.3. Planificación del proyecto

El proyecto ha tenido una duración de ciento cincuenta y nueve días, unas setecientas noventa y cinco horas de trabajo dedicados. Se entiende por elaboración del proyecto la

realización tanto de la aplicación (objeto de este proyecto) como la documentación relacionada con la misma.

Para la realización de este proyecto, se ha seguido el ciclo de vida típico de un proyecto informático con sus fases características y las cuales se exponen a continuación:

1. **Reconocimiento del problema**: Es la primera fase y consiste en saber lo que el cliente quiere, identificar bien el problema y estudiar una serie de soluciones para solventarlo. Es una de las fases más importantes ya que todas las fases siguientes se basan en ella.

2. ***Análisis***: Se extraen los requisitos del producto de software. En esta etapa la habilidad y experiencia en la ingeniería del software es crítica para reconocer requisitos incompletos, ambiguos o contradictorios. Usualmente el cliente/usuario tiene una visión incompleta/inexacta de lo que necesita y es necesario ayudarle para obtener la visión completa de los requerimientos. El contenido de comunicación en esta etapa es muy intenso ya que el objetivo es eliminar la ambigüedad en la medida de lo posible.

3. ***Diseño***: Determinar cómo funcionará de forma general sin entrar en detalles incorporando consideraciones de la implementación tecnológica, como el hardware, la red, etc. Consiste en el diseño de los componentes del sistema que dan respuesta a las funcionalidades descritas en la segunda etapa también conocidas como las *entidades de negocio*. Generalmente se realiza en base a diagramas que permitan describir las interacciones entre las entidades y su secuenciado.

4. ***Desarrollo***: Se traduce el diseño a código. Es la parte más obvia del trabajo de ingeniería de software y la primera en que se obtienen resultados "tangibles". No necesariamente es la etapa más larga ni la más compleja aunque una especificación o diseño incompletos o ambiguos pueden exigir que, tareas propias de las etapas anteriores se tengan que realizarse en esta.

5. ***Pruebas***: Consiste en comprobar que el software responda o realice correctamente las tareas indicadas en la especificación. Es una buena praxis realizar pruebas a distintos niveles (por ejemplo primero a nivel unitario y después de forma integrada de cada componente) y por equipos diferenciados del de desarrollo (pruebas cruzadas entre los programadores o realizadas por un área de test independiente).

6. ***Implantación:*** Es la última etapa antes de hacer entrega del software al cliente. Consiste en desplegar o instalar el software en los equipos a los que tendrá acceso el cliente para su uso al que ha sido diseñado.

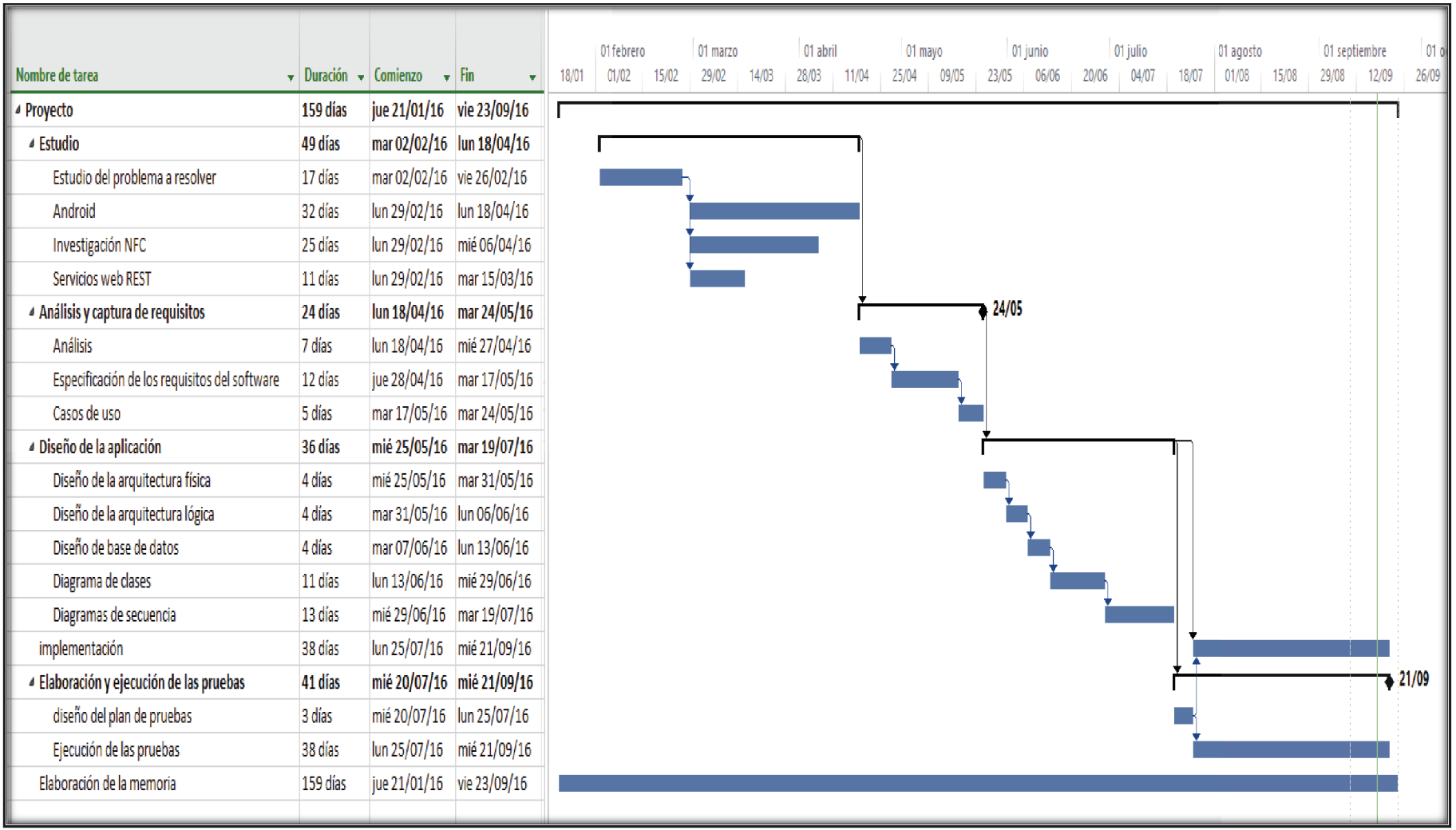
Añadida a estas fases, se puede hablar de otra fase en la que se ha dedicado única y exclusivamente a la realización de esta memoria.

A continuación, en la siguiente tabla se ilustra el tiempo que se ha dedicado cada una de las fases de este proyecto.

FASE	DÍAS	HORAS/DÍA	HORAS TOTALES
Análisis	24	5	120
Diseño	36	5	180
Desarrollo	38	5	190
Pruebas	5	5	25
Implantación	7	5	35
Documentación	49	5	245
TOTAL			***795***

Tabla 51: Reparto horas en fases

7.4. Diagrama de Gantt



Nombre de tarea	Duración	Comienzo	Fin
Proyecto	159 días	jue 21/01/16	vie 23/09/16
Estudio	49 días	mar 02/02/16	lun 18/04/16
Estudio del problema a resolver	17 días	mar 02/02/16	vie 26/02/16
Android	32 días	lun 29/02/16	lun 18/04/16
Investigación NFC	25 días	lun 29/02/16	mié 06/04/16
Servicios web REST	11 días	lun 29/02/16	mar 15/03/16
Análisis y captura de requisitos	24 días	lun 18/04/16	mar 24/05/16
Análisis	7 días	lun 18/04/16	mié 27/04/16
Especificación de los requisitos del software	12 días	jue 28/04/16	mar 17/05/16
Casos de uso	5 días	mar 17/05/16	mar 24/05/16
Diseño de la aplicación	36 días	mié 25/05/16	mar 19/07/16
Diseño de la arquitectura física	4 días	mié 25/05/16	mar 31/05/16
Diseño de la arquitectura lógica	4 días	mar 31/05/16	lun 06/06/16
Diseño de base de datos	4 días	mar 07/06/16	lun 13/06/16
Diagrama de clases	11 días	lun 13/06/16	mié 29/06/16
Diagramas de secuencia	13 días	mié 29/06/16	mar 19/07/16
implementación	38 días	lun 25/07/16	mié 21/09/16
Elaboración y ejecución de las pruebas	41 días	mié 20/07/16	mié 21/09/16
diseño del plan de pruebas	3 días	mié 20/07/16	lun 25/07/16
Ejecución de las pruebas	38 días	lun 25/07/16	mié 21/09/16
Elaboración de la memoria	159 días	jue 21/01/16	vie 23/09/16

Figura 36. Diagrama de Gantt

7.5. Costes

Para el cálculo del coste total de este proyecto se van a tener en cuenta distintos elementos que han intervenido de forma directa o indirecta en la realización del mismo. Al importe total del proyecto se le añadirá un 20% en concepto de costes indirectos tal y como recomienda la plantilla de la universidad.

7.5.1. Coste de personal

Durante la realización del proyecto ha sido necesaria la intervención de los roles de analista, programador y tester.

El analista se encarga de la fase de estudio, diseño y documentar el proyecto, mientras el programador, a partir de la documentación aportada por el analista, codifica los diseños y posteriormente despliega el software en los dispositivos y pone el software disposición del cliente. Para llevar a cabo las pruebas pertinentes que aseguren el funcionamiento del software, se ha introducido el rol del tester que consiste en realizar pruebas exhaustivas dando por válido o no el software desarrollado por el programador.

En la tabla siguiente se va a exponer la carga de trabajo (en horas) de cada uno de los roles así como el coste (en €) de cada uno de ellos y la dedicación en hombre/mes. Se ha considerado que cada mes tiene un total de 40 horas laborables.

Personal	Horas	Precio/Hora	Dedicación(Hombre/mes)	Coste Total €
Analista	300	19	5,25	5700
Programador	600	11,25	2,875	6750
Tester	35	7,5	0,875	262,5
Total				***12.712,50 €***

Tabla 52- Coste personal

Los precios hora utilizados en la tabla se han obtenido a partir de varias ofertas de trabajo consultadas en internet y calculando el equivalente en €/horas a partir del salario anual bruto.

El coste total del personal asciende a 12.712,50€. Este precio no incluye impuestos indirectos.

7.5.2. Coste de Hardware

En el coste del Hardware se incluyen todos aquellos materiales usados para el desarrollo del proyecto y calculando su amortización mediante la fórmula que se muestra a continuación y la cual está incluida en la plantilla que ofrece la universidad como guía para la elaboración del presupuesto.

$$\frac{A}{B} x C x D$$

A: Número de meses desde la fecha de facturación en que el equipo es utilizado.
B: Periodo de depreciación = 60 meses.
C: Coste del equipo.
D: Porcentaje del uso que se le dedica al proyecto.

En la tabla que se expone a continuación se muestran los distintos materiales hardware usados así como el coste de cada uno de ellos aplicándoles la fórmula mencionada arriba.

Descripción	Coste (Euro) sin IVA	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable sin IVA
Toshiba Satellite p50-b-10v	789,21	100	9,00	60	118,38
Ratón inalámbrico Microsoft	22,91	100	9,00	60	3,44
Smartphone Samsung A3	157,21	100	9,00	60	23,58
Placa Arduino Mega	23,62	100	9,00	60	3,54
Tags NFC	3,95	100	5,25	60	0,35
Placa Protoboard + leds + cables cobre	15,79	100	5,25	60	1,38
Impresora HP Laser	62,41	100	0,25	60	0,26

Total	***150,93€***

Tabla 53 Costes por Hardware

Tal y como se puede apreciar en la tabla anterior, el coste por hardware asciende a 150,93 € costes indirectos no incluidos.

7.5.3. Coste de software

En este apartado se recogen todos los costes de software que se han usado durante el desarrollo del proyecto. Como se verá a continuación, la mayoría del software usado es de coste cero.

Software	Coste	Unidades	Coste total
Google drive	0,00	1,00	0,00
Android SDK	0,00	1,00	0,00
Eclipse	0,00	1,00	0,00
GlassFish	0,00	1,00	0,00
Arduino IDE	0,00	1,00	0,00
MySQL	0,00	1,00	0,00
JDBC	0,00	1,00	0,00
Librería Arduino	0,00	1,00	0,00
Microsoft Visio 2013	315,21	1,00	315,21
Microsoft Office 2011	78,21	1,00	78,21
Microsoft Project 2013	607,51	1,00	607,51
TOTAL			***1.000,93€***

Tabla 54 Coste de software

El coste total del software para el proyecto asciende a 1.000,93€ impuestos indirectos no incluidos.

7.5.4. Coste de material fungible

El material fungible es aquel que se ha consumido durante el proyecto y que se ha desgastado durante su uso.

Material	Unidades	Coste en € (sin IVA)	Coste total en € sin IVA
Paquete folios A4	1	1,94	1,94
Tóner impresora	1	54,51	54,51
Lápiz	1	0,59	0,59
Bolígrafo	1	0,59	0,59
Goma de borrar	1	0,47	0,47
Total			***58,10€***

Tabla 55: Material fungible

El coste del material fungible asciende a la cantidad de 58,10€ impuestos indirectos no incluidos.

7.5.5. Coste total

El coste total es la suma de los costes mencionados en los apartados de esta sección, al coste resultante se le añadirá un 20% en concepto de costes indirectos tal y como recomienda la plantilla que se ha usado como guía para realizar este presupuesto.

Concepto	Coste (en €)
Personal	12.712,50
Hardware	150,93
Software	1.000,93
Material fungible	58,1
Costes indirectos (20%)	2784,492
Total	***16.706,95 €***

Tabla 56 Coste total

Y finalmente obtenemos el coste total del proyecto. Todos los costes que se han mencionado no incluyen los impuestos indirectos (IVA).

El coste total asciende a ***16.706.95 €***

A continuación se elabora el presupuesto sobre la plantilla que ofrece la universidad Carlos III como guía.

UNIVERSIDAD CARLOS III DE MADRID
Escuela Politécnica Superior

PRESUPUESTO DE PROYECTO

1.- Autor:
OMAR AZZAHRAOUI
2.- Departamento:
INFORMÁTICA
3.- Descripción del Proyecto:

- Titulo	GRADO EN INGENIERÍA INFORMÁTICA
- Duración (meses)	9
Tasa de costes Indirectos:	20%

4.- Presupuesto total del Proyecto (valores en Euros):
Euros

5.- Desglose presupuestario (costes directos)

PERSONAL

Apellidos y nombre	N.I.F. (no rellenar - solo a titulo informativo)	Categoría	Dedicación [a] (hombres mes)	Coste hombre mes	Coste (Euro)	Firma de conformidad
OMAR AZZAHRAOUI		ANALISTA	5,25	1.085,71	5.699,98	
OMAR AZZAHRAOUI		PROGRAMADOR	2,875	2.348,00	6.750,50	
OMAR AZZAHRAOUI		TESTER	0,875	300,00	262,50	
					0,00	
					0,00	
		Hombres mes 9		**Total**	12.712,98	

[a] 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas)
Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

EQUIPOS

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable [d]
Toshiba Satellite p50-b-10v	789,21	100	9,00	60	118,38
Ratón inhalámbrico Microsoft	22,91	100	9,00	60	3,44
SmartPhone Samsung A3	157,21	100	9,00	60	23,58
Placa Arduino Mega	23,62	100	9,00	60	3,54
Tags NFC	3,95	100	5,25	60	0,35
Placa protoboard + leds + cables	62,41	100	5,25	60	5,46
Impresora HP Laser	15,79	100	0,25	60	0,07
				Total	154,81

[d] Fórmula de cálculo de la Amortización:

$$\frac{A}{B} \text{xCxD}$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado
B = periodo de depreciación (60 meses)
C = coste del equipo (sin IVA)
D = % del uso que se dedica al proyecto (habitualmente 100%)

SUBCONTRATACIÓN DE TAREAS

Descripción	Empresa	Coste imputable
	Total	0,00

OTROS COSTES DIRECTOS DEL PROYECTO[e]

Descripción	Empresa	Costes imputable
Software		1.000,93
Material fungible		58,10
	Total	1.059,03

[e] Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y dietas, otros,...

6.- Resumen de costes

Presupuesto Costes Totales	**Presupuesto Costes Totales**
Personal	12.713
Amortización	155
Subcontratación de tareas	0
Costes de funcionamiento	1.059
Costes Indirectos	2.785
Total	16.712

Figura 37 Hoja de presupuesto total.

El presupuesto total de este proyecto asciende a la cantidad de **DIECISEIS MIL SETECIENTOS SEIS EUROS**.

Madrid a 01 Mayo de 2016

El ingeniero proyectista.

FIRMA

Fdo. Omar Azzahraoui.

Sección 8

8. Conclusiones y trabajo futuro

8.1. Introducción

Finalizada la ejecución del proyecto, se va a proceder a comentar las conclusiones finales a las cuales se han llegado. Además se van a exponer las distintas dificultades a las que se han enfrentado y como finalmente se han resuelto. Para finalizar esta sección se comentarán algunas ideas para futuras mejoras que se podrían realizar sobre este trabajo.

8.2. Conclusiones

El objetivo de este proyecto, tal y como se comentó en secciones anteriores, es aprovechar la tecnología NFC que en la actualidad está de moda para darle un uso diferente al que se le está dando. Un ejemplo de uso de NFC en la actualidad y el cual está teniendo mucho éxito, es el caso de los pagos con el teléfono móvil, esta funcionalidad está revolucionando los mercados y presenta una manera sencilla de realizar compras sin tener que presentar la tarjeta del banco.

La idea, fundamentalmente, se centra en resumir aquellas tareas que se hacen para conseguir adaptar el modo del móvil al sitio en el que uno se encuentre, pues no es lo mismo estar en casa que estar en una reunión.

Lo más habitual es cambiar el modo del teléfono varias veces al día, y la mayoría de las veces se hace por pura necesidad, a modo de ejemplo; cuando una persona está en su casa, para evitar el excesivo consumo de batería o consumo de los datos contratados, deshabilita los datos móviles. Estas tareas normalmente consisten en varios pasos, habrá que acceder a los ajustes del teléfono buscar cada una de las opciones y aplicar cada una de ellas. Además del tiempo que lleva hacer estas tareas, uno tiene que acordarse de todas

ellas, es decir, además de apagar los datos se desea activar la WIFI para tener acceso a internet y poner el volumen en alto entre otras cosas.

La aplicación móvil desarrollada en este proyecto, la parte de cambios de estado, pretende resumir todos los pasos en uno, acercar el móvil a una etiqueta NFC, independientemente del número de tareas que haya por debajo.

Otro uso que se le pretende dar a la tecnología NFC es permitir usar el móvil para realizar tareas cotidianas, una de estas tareas es la de registrar la asistencia al puesto de trabajo. El control de asistencia se realiza en la gran mayoría de las empresas, pues es la única forma de que el usuario pueda demostrar su asistencia a su puesto de trabajo en caso de disputa con la empresa. Esta propuesta pretende eliminar las tarjetas, códigos, firmas, etc. que se aplican en la actualidad y en vez de ello, se acercará el teléfono móvil a una etiqueta NFC y el sistema se encargará de registrar este dato.

Otra tarea propuesta en este proyecto es la del uso de los teléfonos móviles como acreditación y llave para acceso a lugares u objetos que están restringidos, al igual que en el punto anterior, el objetivo de esta propuesta es reducir el uso de tarjetas, códigos, llaves, etc.

Uno de los aspectos más importantes que se debe destacar es que la seguridad absoluta no existe, es decir, siempre se tiene el riesgo de que un empleado le ceda su dispositivo móvil a otro para fichar por él o incluso para dejarle acceder a un sitio restringido bajo su identificación, este caso no se podría detectar y por lo tanto es un punto débil de este proyecto. Sin embargo, todos los métodos usados en la actualidad son vulnerables y presentan el mismo problema. Para intentar mejorar este aspecto se tratará en el apartado de “Trabajos futuros”.

Vista la debilidad de la aplicación, a continuación se comentarán las fortalezas de la misma; en contraposición al punto débil comentado, se ha de explicar que es bastante poco probable que ocurra ya que nadie o prácticamente nadie cede su teléfono a otra persona, esto ocurre ya que los teléfonos son los verdaderos computadores personales y

en los cuales se albergan cantidad de información personal que nadie quiere que otras personas vean (fotos, mensajes, emails, notas, etc.).

Llevar todo en el móvil y olvidarse uno de que tiene que llevarse la cartera encima con tantas tarjetas es un gran alivio, en las carteras o monederos se llevan tantas tarjetas que muchas veces no caben.

Reducir todas aquellas tareas repetitivas para silenciar el móvil, desactivar los datos y el GPS y activar la Wifi en una única tarea es sin duda una facilidad y comodidad para el usuario.

8.3. Dificultades encontradas

Durante la realización de este proyecto me he visto envuelto en multitud de dificultades que me gustaría comentar en este apartado de la memoria y espero sirvan para que se den cuenta del esfuerzo que me ha supuesto para sacar adelante el proyecto fin de grado.

Al inicio de este proyecto y antes de comenzar a trabajar en él, tuve muchos problemas para instalar el entorno, por razones que hasta este mismo momento desconozco, el entorno simplemente dejaba de funcionar sin previo aviso, para solucionar este problema, tuve que salvaguardar todos los datos importantes del equipo y proceder a la reinstalación del sistema operativo del equipo. Realizada esta operación e instalado el entorno nuevamente conseguí que funcionara de manera más estable. Superado este problema, nuevamente detecté que el servidor de aplicaciones GlassFish, en ocasiones, tardaba mucho en arrancar e incluso a veces no arrancaba dando errores, tras analizar estos errores, he detectado que se trata de incompatibilidad entre la versión de GlassFish y la máquina virtual de Java, inmediatamente actualicé la máquina y el error se resolvió.

Sin duda la mayor dificultad que se me presentó fue poder leer y escribir mediante NFC en una etiqueta, fue tal la dificultad que me llevó varias semanas dar con la solución. Este error sucedía (sólo en algunas ocasiones) al acercar el móvil a una etiqueta NFC para leer o guardar y consistía en que la aplicación móvil se cerraba inesperadamente y sin previo

aviso, para solventar este error tuve que aprender el mecanismo de control y uso de NFC en Android cuyo contenido encontré en la página oficial y varios foros, pues al parecer había que tener en cuenta algunos parámetros que yo evidentemente no los tenía en cuenta.

Cabe destacar que dado que no tenía mucha idea sobre la programación en Android, al principio fue muy complejo empezar y una vez fui adquiriendo experiencia fui mejorando la aplicación hasta conseguir una aplicación funcionalmente estable.

En general todas las dificultades técnicas fueron resultas satisfactoriamente haciendo uso en ocasiones de la documentación oficial y otras ojeando foros de internet e incluso seguir algunos video tutoriales encontrados en YouTube.

Algunas funciones en Android precisan de que el usuario intervenga directamente para habilitarlas o deshabilitarlas, estas funciones afectan sobre todo a la funcionalidad de cambio de estados. Para aplicar un estado en el que se tiene definido habilitar o deshabilitar cualquiera de las siguientes opciones requiere que el usuario lo haga de forma manual en los ajustes del teléfono:

- GPS.
- Datos móviles.
- Modo avión.

Para facilitar la tarea al usuario, cuando haya alguna de estas opciones, la aplicación abre los ajustes y ofrece al usuario la interfaz para habilitar o deshabilitar dicha opción.

Estas restricciones son para evitar que aplicaciones de terceros hagan un uso indebido de estas opciones sin que éste lo sepa.

Durante el planteamiento del proyecto, se propuso hacer el cambio de estados del teléfono sin que el usuario tenga que desbloquear la pantalla del teléfono, sin embargo, hay puesta una restricción en Android de forma que sólo se puede usar NFC en un Activity –clase

con interfaz gráfica- lo que supone que no se puede usar desde un proceso que se ejecuta en segundo plano y esto implica que el usuario, cuando quiera usar la aplicación, tiene que desbloquear la pantalla del teléfono si está bloqueada.

8.4. Trabajo futuro

En este apartado se expondrán algunas ideas que por razones diversas no se han llegado a implementar y que sin duda suponen mejoras sustanciales para la aplicación.

En primer lugar, para facilitarles la vida a los administradores a la hora de registrar un dispositivo y enlazarlo con una cuenta de usuario, en vez de introducir el código generado de forma manual, se propone generar este código en forma QR. Para añadir este código sólo habría que activar la opción correspondiente y el móvil tomará una instantánea del código QR reconociendo el código que contiene.

Como segunda mejora se propone mejorar la interfaz gráfica ya que tal y como está hecho el trabajo ahora mismo es bastante sencillo.

Se propone también cifrar las comunicaciones entre el cliente (aplicación Android) y el servidor (aplicación alojada en GlassFish) de extremo a extremo y usar Tokens temporales para realizar dicho cifrado.

Con el fin de mejorar la autenticación del cliente y evitar fraude en el uso de la aplicación para realizar el control de presencia o de acceso, se propone implementar métodos de autenticación fiables tales como el Authenticator de Google, lector de huella dactilar (disponible en algunos terminales Android), etc.

Perder el móvil es sin duda una de las mayores faenas y lamentablemente sucede con bastante frecuencia y por ello surge la necesidad de llevar a cabo un sistema para mantener los datos relacionados con los estados creados en la nube, de esta forma se garantiza que se puedan recuperar en caso de pérdida del móvil o caso en el que se tenga más de uno.

Una de las ideas que inicialmente se quería implementar y que por falta de tiempo no se ha llegado a realizar es la de autenticación del usuario en los ordenadores de las aulas de informática de una universidad, biblioteca, etc. Sería necesario investigar la forma y la tecnología a utilizar para que de una forma centralizada se permita el acceso a los equipos sin que el usuario tenga que introducir el usuario y la contraseña.

Glosario de términos

Término	Significado
Bluetooth	Tecnología de comunicaciones de corto alcance usado para la unificación y comunicación entre distintos dispositivos.
GPS	Global Positioning System. Es un sistema de posicionamiento que hace uso de satélites para detectar la posición de los objetos.
NFC	Near Field Comunication: comunicación de campo cercano. Tecnología de comunicaciones de corto alcance.
RFID	Radio Frequency Identification. Sistema de almacenamiento y recuperación remoto de datos mediante el uso de la radio frecuencia.
Wifi	Tecnología de comunicaciones inalámbrica comúnmente usada para interconectar equipos en redes locales en unas distancias que oscilan entre veinte y trescientos metros de distancia.
Activity	Terminología Android para describir una clase especial usada para representar la parte gráfica de la aplicación.
SOAP	Simple Object Access Protocol, es un protocolo estándar que define el intercambio de un objeto entre distintas plataformas mediante el uso de ficheros XML.
REST	Representational State Transfer, es un estilo de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web.
Token	Es un código único generado en un sistema con el fin de identificar a los usuarios conectados a un servidor.

Tabla 57. Glosario de términos

Referencias

[1] Hipertextual, Motorola DynaTAC: primer teléfono móvil. http://hipertextual.com/2014/03/motorola-dynatac-30-aniversario-venta

[2] Apple, Using Apple Pay in stores and within apps. https://support.apple.com/en-us/HT201239

[3] Apple, Apple Developer. https://developer.apple.com

[4] Google, Android Developers. http://developer.android.com

[5] Jesús Tomás Girones, El gran libro de Android avanzado. Marcombo, Sep 24, 2013.

[6] M. Kerschberger, Near Field Communication. A survey of safety and security measures. Vienna, July 17, 2011.

[7] E. V. García, Desarrollo de una aplicación de control de acceso y sistemas de identificación mediante la tecnología NFC. Universidad Carlos III de Madrid, 2011.

[8] RFID point, Cuál es el origen de la tecnología RFID? http://www.rfidpoint.com

[9] Wikipedia, RFID. http://es.wikipedia.org/wiki/RFID.

[10] M. F. Carignano, P. Ferreyra, Tecnología inalámbrica Near Field Communication y sus aplicaciones en sistemas embebidos. Congreso argentino de sistemas embebidos (CASE), 2011.

[11] G. Chavira, S. W. Nava, R. Hervás, J. Bravo, Carlos Sánchez, Localización e Identificación: Una combinación RFID-NFC.

[12] L. E Ortiz Fernández, Diseño e implementación del prototipo de un dispositivo identificador de objetos de uso común para personas no videntes basado en la tecnología RFID. Universidad Politécnica Salesiana, 2012.

[13] A. Campa Ruiz, Desarrollo de una aplicación de pago a través de la tecnología NFC. Universidad Carlos III de Madrid, 2011.

[14] L. Tolsada Bris, Desarrollo de una aplicación de transferencia de ficheros basada en NFC y Bluetooth. Universidad Carlos III de Madrid, 2012

[15] Blogspot, Ventajas y desventajas del RFID. http://anita315.blogspot.com/2005/10/ventajas-y-desventajas-del-rfid.html

[16] D. I. Tapia, J. R. Cueli y varios autores, Identificación por Radiofrecuencia: Fundamentos y Aplicaciones. Las Jornadas Científicas sobre RFID, 2007.

[17] D. A. Chavarría, TECNOLOGÍA DE COMUNICACIÓN DE CAMPO CERCANO (NFC) Y SUS APLICACIONES. Universidad de Costa Rica, 2011.

[18] M. V. Bueno Delgado, P. Pavón Mariño, A. de Gea García, La tecnología NFC y sus aplicaciones en un entorno universitario. Departamento de Tecnologías de la Información y las Comunicaciones, Universidad Politécnica de Cartagena, 2011.

[19] Terra, NFC Forum. http://www.terra.es/personal/ccossio/tecnologiaNFC 9.htm

[20] J. Bravo, C. Sánchez y varios autores, La NFC: una nueva forma de concebir la RFID. Aplicación para grandes superficies, Artículo para las jornadas de RFID, 2007.

[21] RFID MAGAZINE, Etiquetas NFC. http://www.terra.es/personal/ccossio/tecnologiaNFC 7.htm

[22] D. F. Veloz, diseño e implementación de un prototipo para el control de acceso de personas aplicando la tecnología NFC por medio del uso de teléfonos celulares compatibles con esta tecnología. Escuela Politécnica Nacional, 2010.

[23] F. Gallego de la Sacristana, Aplicación de inicio de sesión mediante autenticación con NFC. Universidad Carlos III de Madrid, 2011.

[24] J. Areitio Bertolín, Análisis de los riesgos y contramedidas en seguridad-privacidad de la tecnología NFC en móviles. Seguridad en redes, 2011.

[25] 16. Fielding, Roy Thomas. Architectural Styles and the Design of Network-based Software Architectures. [En línea] 2000. https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf.

[26] Marset, Rafael Navarro. REST vs Web Services. [En línea] 2006-2007. http://users.dsic.upv.es/~rnavarro/NewWeb/docs/RestVsWebServices.pdf.

[27] Wladimir Rodríguez - Universidad de Los Andes - Mérida (Venezuela)-. Arquitectura orientada a servicios. Videos youtube (8 videos). [En línea] 2013. https://www.youtube.com/watch?v=UJ3ZVoQitmw&index=1&list=PLcL8RDzOxvIrTVV-7G1TkNeyBc9oFmmhg.

[28] E. Vilar – periódico La Razón – Madrid. Nomofobia: La enfermedad que quizás padece y no lo sabe. [En línea] 2012. http://www.larazon.es/historico/6785-nomofobia-la-enfermedad-que-quizas-padece-y-no-lo-sabe-MLLA_RAZON_436328#.Ttt1ivov9EIgG1c

SUMMARY

Project description

The project consists in create a mobile application that is capable to modify automatically phone states and it is based on information obtained through NFC tags. Each tag will store a code that will refer to a particular state. To obtain this information from an NFC tag it will be necessary the first of all to save the tag. The application defines this process as a definition of modes, available as an option on the application. The user, first, has to select the appropriate option of mode definitions, and then will choose the audio mode and the possibility of activating other options, such as WiFi, Bluetooth, GPS, mobile data module, etc. The user, once the mode is defined, will select the appropriate option to save this data in an NFC tag approaching the device to the tag.

Therefore, this project aims to facilitate the task of changing phone status, so that the user only has to perform a single task: to bring the phone near to an NFC tag, instead of attending several steps to finally get applied the required mode.

Some project proposals to make more widespread the use of NFC technology are: use mobile to record the entry and exit of work; and use it as a key for access to restricted areas or to open a safe. The final proposal is presented as two new modules called "Presence control" and "Access Control".

The Presence control module identifies the company personnel and records the date and time in the database. As well as the GPS coordinates where the people is at the time of that registration. This process is carried out in almost all companies in order to have more control over their workers.

The mechanisms used today, such as band cards, smart cards, signature on paper, etc., are unreliable mechanisms because it is possible giving the card to someone else or signing for other person what it is the same that commit fraud. However, the method proposed here is reliable for several reasons: firstly, the worker cannot sign unless he is very close

to the NFC tag, which contains the information needed to complete the process. On the other hand, it is unlikely that someone leave its mobile device to other people due to the dependence on the mobile phone and because it may contain sensitive and personal information. However, there is always the risk that a person give the device to another person in order to perform the process. To minimize this risk, see future work section.

The "Access Control" option is to use the mobile as a key to access restricted areas of a building or any object locked away. This key will send a unique code to the server and it will handle opening the door allowing access to the person that has the device but only if he has the permissions needed. The process is similar to the above, i.e., the person put the phone close to an NFC tag, the APP will send the request to the server and it will respond depending on whether this person has or no access, also the server will send a signal to the door mechanism to allow access in case the permissions are available.

In general, the use of cards has the drawback of accumulating them reaching the point of not having room for more. Even, there is a risk of losing them very easily. So that, this project aims to unify all in one via mobile device.

The main objectives of this project are:

Create an application capable to modify phone state automatically and only based on the information contained in the NFC tags.

The application will be able to save the user-defined modes in NFC tags.

With the screen device unlocked, it may be applied the mode that contain a label so that it were not necessary to start the application.

The application will allow signing (recording date, time, GPS coordinates and what user does) without entering pin, passwords, etc., by reading only an enabled tag for that purpose and connecting remotely to a server.

Using the device as a key to access restricted areas. This process will be similar to the control of presence or signing, with the difference that in this case, the server will send information to the device of the door lock to indicate or give the order for open it. In the project, this process will be simulated using an Arduino.

Personal goals: To carry out this project, I proposed a series of personal goals that I will describe. These objectives will help to make this project and, they will provide knowledge and experience to my future career.

The study in-depth of language JAVA.

Web services and especially those that are based on REST are the key to the realization of the control presence and access control. To achieve interconnect two applications whose platforms are different it will be necessary further study about this technology.

Study and learning of communications technology NFC nearby fields, currently I am completely unaware of the use and implementation of systems based on it, so I plan to make a theoretical and practical research on this technology.

Since the application will be developed for the Android operating system and since I have no experience in deploying applications for these systems, I plan to do an intensive course to carry out this Project.

State of the art

Since the appearance of the first mobile phone, the need to silence the phone arose, not only remove the sound, but also it happened the vibrator invention for notified in other much more discreetly way. In fact, the vibrator was invented by Motorola in 1984, i.e., with the launch of the first phones to the market.

The rise of mobile telephony and, above all, the advent of Smart Phones and the inclusion of new technologies in it, has become necessary management and administration. A

second-generation phone (nineties) allowed selecting various sound modes, actual phones are virtually 100% configurable.

To enable or disable the wireless mobile module, for example, or to change the status of the phone to silent mode, are not difficult tasks to perform. However, if you have multiple tasks combined that you have to make repeatedly during a day they will be tedious tasks. In addition to changing the sound mode, sometimes you are interested also in changing the Wi-Fi mode, Bluetooth mode, GPS... These needs arise to avoid disturbing others or simply to save battery mobile, because one of the main problems of mobile in these days is the low battery life.

In the newest phones, third and fourth generation phones, manufacturers of operating systems and mobile software developer community, carry out improvements to make these tasks are the least tedious as possible, i.e., having the scope and try that with only one click the phone sound status change and also turn off the Wi-Fi and Bluetooth modules among others.

The latest versions of Android and IOS operating systems, allow access from the main screen to a widget, where in an easy and quick way you can enable or disable several functions that are available for the terminal.

Today, the phone has gone from being a device to call and send SMS messages exclusively to be a powerful computer with capabilities that sometimes exceed some desktops or laptops with a few years of life.

Conduct a presence control, also called "transfer in and out" within companies, or what is the same: record the time of entry and exit to the job. This is almost essential in most of them, and each time more advanced technologies are used. In many companies, this operation can significantly vary the wage of a worker in case it is not done properly, because, in many cases, it can be understood as a lack of punctuality and, therefore, not meeting the agreement with the company.

Depending on the organization, you can find several ways to make that record. In some of them, nowadays, you need to sign a document in the presence of another responsible person. This practice is too old, but persists despite of the great technological advances that has been reached. This is so by the simple fact that the company understands that it is a safe way that they have to control that the worker complies with its obligations. Companies that have advanced in this area are using smart cards, fingerprints or secret codes together with the person ID that sign. All with the aim of carrying out an exhaustive control on compliance with the working hours of workers.

As the need that organizations have to carry out checks on compliance with punctuality workers, also it arises the need to reserve the right of access to certain sites for people with prior authorization. In most cases, these sites are sensitive because of the contained material or stored information in them. Set access to certain areas by responsibility degree is a common practice in many companies and public institutions.

The access to several facilities of an organization, like signing in and out, is one of the most common practices that arise from the need to control these facilities in order to protect and safeguard everything that are considered as sensitive. Therefore, it requires that people that can access was previously authorized. Smart cards, keys (the usual), porters with access code and fingerprints are some of the many forms that are used to allow or deny access, however, as in the case of the signings we have seen before, it requires carry cards or keys, remember the pin, and so on.

System analysis

This project consists in develop, on one hand, a mobile application and, on the other, a server application that simply will host a web service that will respond to the requests made by the mobile application. Part of the mobile application will be independent of the web service; however, the presence and access control parts do not work without access to the web service.

One of the features of the mobile application is to change the phone's status based on NFC tags. This functionality performs locally and completely autonomously by the mobile

application and it is available to all users. The access control and control of presence features, however, require a connection to the server to check if the mobile device has permission to perform the requested operation. Each of the operations (control of presence and access control) have separate permissions between them, i.e., a device can have permissions to connect to the web service for control of presence and not have permission to access control and vice versa, or it can also occur that it have the permissions to perform both operations. To identify users with their mobile device, the user may request a link to an account, to do this, the user will have to go personally to the department of his company and the personnel responsible for registration and control staff will assign an account and will link the user device with that account. Authorized personnel can only perform this operation and the user cannot change this link.

In order to carry out the operation for changing phone states, the user needs an NFC tag that contains the unique code of the phone mode that he wants to apply, the NFC Interactive application provides functionality to define different phone modes and save them in NFC tags. Each label will contain the code for a single state. Once you have the NFC label with the mode code, he only have to put the mobile with the NFC function enabled and with the screen unlocked close to the tag.

For presence control, as in the case of change of state, it will need an NFC tag with a unique code of the company generated in the system and stored in the database. In order to store the code I have decided to use external applications because the operation consists only to store a code in an NFC tag and to carry out this operation, there are many applications, but it is not discarded that in the future someone develops an explicitly application for this purpose. First, you must have the NFC tag with the company code, and then the procedure is as follows. The mobile will approach to the NFC tag with NFC, GPS enabled, and the screen unlocked. Then the mobile application will send to the server the unique code of the mobile's link with the user account. Finally, the GPS coordinates of the place where the person is at the time. The server will verify the data and will store in the database the sign of the user whose account is linked to the mobile that he used. In the case of access control, it will be similar to the control of presence. In the case of presence control, the server, once the operation is completed, send a response message

either favorable or unfavorable. In the case of access control is also send the signal to open the door allowing access in case that it corresponds.

Future work

In this section, some ideas that for various reasons are not implemented yet and that undoubtedly represent substantial improvements to the application, will be presented.

First, to make life easier for administrators when registering a device and linking it with a user account, instead of entering the generated code manually, it is proposed to generate that code in the QR code form. To add this code, it would only have to activate the appropriate option and the mobile will take a snapshot of the QR code recognizing the code it contains.

A second improvement should be to improve the graphical interface because as it has done right now is quite simple.

It is also proposed to encrypt communications between the client (Android application) and server (GlassFish hosted application) from end to end and to use temporary tokens to perform such encryption.

In order to improve client authentication and prevent fraud in the use of the application for presence or access control, it proposes to implement reliable methods of authentication such as the Google Authenticator, fingerprint reader (available on some Android terminals), etc.

To lose the mobile is undoubtedly one of the major damage and unfortunately it happens quite frequently and therefore the need arises to carry out a system to keep data related to created states in the cloud, this ensures that they can be recovered in case that you lose the mobile or in case that you have more than one.

One of the ideas that initially I should want to implement and, because of that lack of time, it has not come to realize, is the user authentication on the computers of the

computer rooms of universities, libraries, etc. It would be necessary to investigate technology that uses in a centralized way allow access to computers without the need for users to enter username and password.

Conclusions

The objective of this project, as discussed in previous sections, is to use the NFC technology, that is currently fashionable, and give it a different use. An example of use of NFC nowadays that is very successful, is the case of payments with mobile phone, this functionality is revolutionizing markets and presents an easy way to make purchases without having to bring the bankcard.

The idea, basically, focuses on summarizing those tasks that are made for adapting mode mobile to the place where one is located, it is not the same to be at home than in a meeting.

The most common is to change the phone mode several times a day, and most of the times it is done out of necessity, for example; when a person is at home, to avoid excessive battery or contracted data consumption, he can disable mobile data. These tasks usually consist of several steps: to access phone settings, find each of the options and apply each one. In addition to the time it takes to do these tasks, you have to remember all of them, i.e., besides to turn off the data you want to activate the Wi-Fi to have access to Internet and put the volume up among other things.

The mobile application developed in this project, the state changes part, intended to summarize all the steps in one, close the phone to an NFC tag, regardless of the number of tasks that are behind.

Another use that is intended to give the NFC technology is to allow mobile the use of the mobile to perform daily task, one of these tasks is to record attendance at the workplace. Most of the companies carry out attendance control; it is the only way that the user can show his support to his job in case of dispute with the company. This proposal aims to remove cards, codes, signatures, etc. that are apply nowadays and instead, the mobile phone will close to an NFC tag and the system will register this data.

Another task proposed in this project is the use of mobile phones as accreditation and key for access to places or objects that are restricted, as in the previous point, the objective of this proposal is to reduce the use of cards, codes, keys, etc.

One of the most important aspects that should highlight is that absolute security does not exist. The risk that an employee give up their mobile device to another employee to sign for him or, even, to let access to a restricted site under identification always exists. This case could not be detected and, therefore, it is a weak point of this project. However, all the methods currently used are vulnerable and have the same problem. In order to try to improve this aspect were discussed in the section "Future work".

Having regard to the weakness of the application, the strengths of it will be discussed now. As opposed to weakness commented, it has to explain that it is quite unlikely that it occur since no one or almost no one gives his phone to someone else, because the phones are true personal computers in which people store personal information that does not want other people to see (photos, messages, emails, notes, etc.).

Carry everything in the mobile and forget that one has to take the portfolio in with many cards is a great relief. In wallets or purses, people take many cards that often do not fit.

Reduce all those repetitive tasks to silence mobile, disable data or GPS and activate the Wi-Fi in a single task is definitely an easy and comfort thing for the user.